MTL TR 89-36

AD

AD-A208 755

AUTOMATED IMMERSION TESTING ROBOTIC SYSTEM - HARDWARE AND SOFTWARE DESIGN

SUZANNE G. W. DUNN
POLYMER RESEARCH BRANCH

April 1989

Approved for public release; distribution unlimited.

US ARMY
LABORATORY COMMAND
MATERIALS TECHNOLOGY LABORATORY

U.S. ARMY MATERIALS TECHNOLOGY LABORATORY
Watertown, Massachusetts 02172-0001

89 6 08 093

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: MTL TR 89-36	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): AUTOMATED IMMERSION TESTING ROBOTIC SYSTEM - HARDWARE AND SOFTWARE DESIGN	5. TYPE OF REPORT & PERIOD COVERED: Final Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Suzanne G. W. Dunn	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: U.S. Army Materials Technology Laboratory, Watertown, Massachusetts 02172-0001, SLCMT-EMP	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: D/A Project: 1L162105.AH84
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Laboratory Command, 2800 Powder Mill Road, Adelphi, Maryland 20783-1145	12. REPORT DATE: April 1989
	13. NUMBER OF PAGES: 59
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Laboratory automation	Composites	Polymers
Immersion testing	Materials durability	Liquid diffusion
Robotics	Environmental testing	Moisture uptake

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(SEE REVERSE SIDE)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Block No. 20

ABSTRACT

Hardware and software development of a fully automated robotic work cell to determine the water sorption-diffusion properties of plastics and organic matrix composite materials and to handle all data acquisition is described. Immersion testing involves measuring weight changes of specimens as a function of the time they are immersed in water at a particular temperature. This test method is labor intensive and tedious, involving repetitive operations such as transferring liquids and specimens, blotting and weighing specimens, and recording data. Immersion studies can take years to complete and generate enormous amounts of data which must be recorded and evaluated. Through automation, immersion testing is no longer limited to an 8-hour work schedule, labor costs have been reduced, productivity has been increased tenfold, and the reliability, precision, and accuracy of the test have been improved.

CONTENTS

I. INTRODUCTION

The moisture content of fiber-reinforced polymer matrix composite materials significantly affects their durability and mechanical properties. Immersion testing, used to investigate the moisture sorption-diffusion behavior of composites, involves measuring weight changes of specimens as a function of the time they are immersed in water at a particular temperature. Immersion testing is tedious and time consuming. Depending upon the characteristics of the composite and temperature, a single test may run for several months, generating large amounts of data which must be collected and evaluated. Weight measurements are limited to an 8-hour work schedule, and are subject to operator errors. To successfully characterize the sorption-diffusion properties of composites, many immersion tests under various conditions are required. Manpower constraints limit the number and variety of conditions at which materials can be evaluated.

A fully automated robotic system has been developed for performing immersion studies in support of polymer research to increase the productivity for the evaluation of existing and next generation materials. Our goals in designing the automated immersion testing system were: (1) to emulate or improve upon manual operations, (2) to increase sample throughput, (3) to improve the precision and accuracy of measurements, and (4) to provide sufficient flexibility to handle different types of specimens and enable future expansion of operations.

State-of-the-art laboratory robotics coupled with today's personal computers provides a flexible means for addressing the problems associated with immersion testing procedures in the laboratory. In the following sections, this report describes the details of the immersion test method and the hardware and software design of the robotics work cell developed for performing automated immersion studies continuously and unattended. Section II (Analytical Procedure) describes the test method for performing manual and automated immersion studies. Section III (The Analysis of Moisture Sorption Data) discusses the method for interpreting immersion test data and how this information may be used in the evaluation of composite materials. Our objectives and design constraints, including the anticipated problems and concerns with maintaining the integrity of an immersion study using laboratory robotics, are discussed in Section IV (Design Objectives and Constraints). The hardware and software designs are detailed in Sections V (Hardware Design) and VI (Software Design), respectively. Section VII (System Verification) discusses the studies performed to compare the manual and automated test methods for validating the Automated Immersion Testing Robotic System, and illustrates the significant improvements achieved in the immersion test method with the use of laboratory robotics. Current plans for improving the design, and extending the immersion testing system to include accelerated aging studies in efforts to evaluate the durability and predict the lifecycle of composite materials, are discussed in Section VIII (Future Enhancements).

II. ANALYTICAL PROCEDURE

To fully understand the moisture sorption-diffusion behavior of one composite material processed under one set of processing parameters, numerous immersion tests under various temperature conditions are required. A single immersion test involves making periodic weight measurements and recording the change in the specimen's weight, with respect to the specimen's initial weight, as a function of immersion time. The following paragraphs describe the manual and automated immersion test method procedures, respectively.

Manual Test Method

Organic matrix composite test specimens are typically conditioned by drying under vacuum at 60°C (or below their glass transition temperature) for a period of 48 hours (or until there is no further decrease in weight). After allowing the specimens to equilibrate in a desiccator, their dimensions are measured and they are transferred into sample jars which contain 60 ml of the immersion liquid (often at elevated temperatures). Weight measurements are taken after the specimens have been immersed for 30 minutes, 1 hour, 2 hours, 4 hours, 6 hours, 24 hours, and 30 hours. Subsequent weight measurements are taken weekly until the specimens no longer exhibit a change in weight.

The weighing process involves removing the specimen from the immersion liquid, blotting the specimen with high absorbent filter paper to remove all surface liquid droplets, weighing the specimen, and returning it to the immersion liquid as quickly as possible. For each weight measurement, the total time the specimen has been immersed and the actual weight reading is recorded. By plotting the percent weight change versus the square root of time, the diffusion rate may be calculated using the Fickian law of diffusion, described in Section III (Analysis of Moisture Sorption Data).

Automated Test Method

In the initial designing phase of the immersion testing robotic work cell, the primary goal was to develop a system that performed immersion tests precisely the same way as in the established manual immersion testing procedures. Although the immersion test method is straightforward, many complex operations are required; e.g., manipulating small specimens, capping and uncapping specimen jars, blotting the surface liquid droplets from specimens, and making accurate weight measurements.

Initializing a specimen into the Automated Immersion Testing Robotic System involves manual as well as automated operations. First, specimens are conditioned manually (as stated in the Manual Test Method Section), their dimensional measurements are taken and entered into the computer by the operator, and the specimen is placed in a specimen holder within the robot's work envelope. The robot then proceeds by taking the specimen's "initial" weight measurement, placing the specimen in a designated specimen jar containing the immersion liquid at a given temperature, and returning the specimen jar to a temperature-controlled circulating water bath.

The process of making weight measurements for the remainder of the immersion test procedure follows the manual test method operations, as stated previously, with two major exceptions. First, the robot performs the required operations, and the computer records and calculates all immersion test data. Second, the robot is allowed to make weight measurements more frequently during the normal 8-hour work day, overnight, and on weekends.

III. ANALYSIS OF MOISTURE SORPTION DATA

The objective of the immersion test is to determine the moisture content (percent weight gain) of a material as a function of its immersion time. When water is used as the immersion liquid, the test is essentially the ASTM Standard Test Method for Water Absorption of Plastics (D 570).[1] Determinations of the relative rate of water absorption are important in evaluating the effects of moisture exposure on such properties as mechanical strength, electrical resistivity, dielectric losses, physical dimensions, and appearance. Moisture content has significant effects on physical-mechanical properties and the long-term durability of polymeric and organic matrix composite materials.

To interpret immersion test data, moisture diffusion through the thickness of a test specimen can be described using a one-dimensional Fickian equation

$$dC/dt = D\ d^2C/dx^2 \tag{1}$$

where C is the moisture concentration in the specimen, x is the thickness parameter, t is exposure time, and D is the effective diffusivity of moisture through the thickness. Assuming that moisture diffusion and temperature are constant inside the specimen and under appropriate boundary conditions,[2] the solution for Equation 1 can be approximated as

$$C_i = G\ (C_e - C_o) + C_o \tag{2}$$

where C_i is the moisture content of the specimen at time t_i, C_e is the moisture content of the specimen at equilibrium (maximum moisture uptake), C_o is the initial moisture content (assumed to be uniform or zero), G is a time-dependent parameter defined as

$$G_i = 1 - \exp\left[-7.30\left(\frac{Dt_i}{h^2}\right)^{0.75}\right], \tag{3}$$

and h is the thickness of the specimen exposed on both sides. Since (Equation 2) the percentage weight (wt%) gain $W_i = 100\ G_i$, moisture diffusivity D_o can be determined directly from experimental data by plotting W_i versus $(t_i)^{1/2}$ and calculated either from the initial slope S_o

$$D_o = 0.196 \bullet h^2 \bullet (S_o/W_e)^2 \tag{4}$$

or from the time $t_{1/2}$ corresponding to one-half of the maximum moisture uptake value W_e for the specimen, as illustrated in Figure 1,

$$D_o = 0.049 \bullet (h^2/t_{1/2})\ . \tag{5}$$

The diffusivity D_o and equilibrium moisture content W_e are parameters which describe a specimen's moisture diffusion properties.

Figure 1 illustrates the typical sorption-diffusion curve following the Fickian law of diffusion.

1. ASTM D 570. *Standard Test Method for Water Absorption of Plastics.* 1987 Annual Book of ASTM Standards, v. 08.01, American Society for Testing and Materials, Philadelphia, PA, 1987.
2. SHEN, C-W, and SPRINGER, G. S. *Moisture Absorption and Desorption of Composite Materials.* J. Composite Materials, v. 10, 1976, p. 2-20.

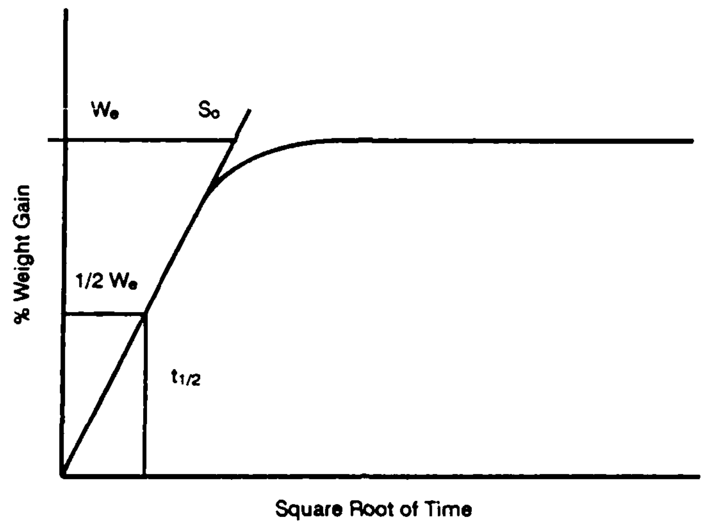

Figure 1. Typical moisture sorption/diffusion curve.

IV. DESIGN OBJECTIVES AND CONSTRAINTS

In addition to a flexible, programmable robot, the automation of immersion testing required peripheral equipment for: (1) maintaining specimen jars, immersion liquids, and test specimens at a constant and elevated temperature, (2) capping and uncapping specimen jars, (3) blotting specimens, and (4) making accurate weight measurements to the fifth decimal place. In the process of selecting and/or designing this specialized equipment, we found that the system could not be designed easily and reliably to handle all immersion tests in a generic sense. For this reason constraints were established.

System Constraints

The immersion testing robotic work cell was designed around two constraints; specimen configuration and limitations on immersion liquids.

Specimen Configuration: Since immersion specimens vary in size, shape, thickness, rigidity, and uniformity, standardized sample specifications had to be established. In this initial work, it was therefore decided to restrict the test to rigid, one-inch square or disk specimens of relatively uniform thickness (0.5 mm-1.9 mm). This decision was made for three reasons: (1) there is a high and continuing demand for evaluating composite materials; composite specimens are rigid and easily fabricated to the specified dimensions, (2) the specimen size is relatively consistent with ASTM test methods, and (3) uniform specimens of this configuration allow the robot the ability to handle specimens directly with custom forcepslike fingers, and allow the blotting operations to be simplified for enhanced reliability.

Immersion Liquids: Liquids used in immersion testing also vary. Some liquids are hazardous or corrosive, requiring specimens to be self-contained in closed jars. The majority of immersion tests, especially for testing composites, are performed with water or saline solutions. To provide as much flexibility as possible, a decision was made to demonstrate the system using "safe" liquids, primarily distilled water, while simulating more hazardous situations by using screw-capped containers for future adaptations to more hazardous environments.

Design Objectives

As mentioned in the Introduction, our goals in designing the immersion testing system were: (1) to emulate or improve upon operations as specified in the manual immersion test method, (2) to increase sample throughput, (3) to improve the precision and accuracy of measurements, and (4) to provide sufficient flexibility to handle different types of specimens and enable future expansion of operations. Although there were many anticipated advantages to automating the immersion test procedure, there were also many concerns with maintaining the integrity of the test method. The following paragraphs address our design goals in greater depth, and point out advantageous as well as potential problem areas.

Improved Operations: Although the automated immersion test method duplicates the mechanics of the manual test method, several improvements were made in the test procedure. First, as discussed in Section II (Analytical Procedure), the manual test method limited weight measurements to five measurements on day 1, two measurements on day 2, one measurement on day 3, and one measurement per week for the remainder of the study. Through automation, weight measurements may be taken more frequently during a normal working schedule, and additional information may be obtained by making measurements overnight and during weekends and holidays. Using a simple algorithm, the robot can be programmed to look at past data to determine and alter the time interval between weight measurements as necessary.

One of the anticipated advantages to making more frequent weight measurements was to be able to capture information anomalous to Fickian's law of diffusion. Along with this expected advantage came a potential problem. Since the number of weight measurements taken during the course of an immersion study was significantly increased, we were concerned about the affects that this might have on the overall integrity of immersion test data (i.e., whether the increased amount of time that the specimens are removed from their respective liquid environments would affect overall test results).

The second major improvement in operations concerns specimen handling. In the manual test method, specimens are allowed to lay flat on the bottom of their respective sample jars, limiting the flow of immersion liquid around the specimens. To address this problem, and to provide a means for the robot to manipulate specimens into and out of specimen jars, sample jars were custom designed to hold rigid specimens vertically (refer to Section V, Hardware Design).

Sample Throughput: Although automation did not decrease the amount of time required to perform an immersion test, many more specimens may be evaluated during a period of time using robotics. Materials may now be evaluated under a variety of temperature conditions. Materials processed under varying processing conditions may now be studied. Along with performing the mechanical tasks of immersion testing, data handling, too, affects sample throughput. Robotics, coupled with computers, make the time required for data acquisition, data handling, and data reduction negligible.

Robotics generally include feedback and control systems which verify accuracy and functionality for each operation. For this reason, the automated system takes twice as long to make a single weight measurement as a human operator. Determining what effects this additional length of time would have on the overall test result was a major concern.

Precision and Accuracy of Measurements: A large advantage of using robotics is that each weight measurement for each specimen is performed in exactly the same way. Measurements are taken exactly on time, and data acquisition is not subject to human error.

One major concern in maintaining and/or improving the accuracy of measurements was in the specimen blotting procedure. The manual blotting process relies on visual inspection to ensure that all water droplets have been removed from the specimen prior to a weight measurement (i.e., one droplet left on a specimen will result in a significant increase in weight). Since the automated system does not have that visual capability, the concern was in the reliability of removing all droplets from all surfaces, including specimen edges.

Flexibility: It was important to limit the scope of our design initially in order to produce a successful design with a limited cost investment. However, anticipating future applications, the robotic system was designed with the flexibility required to expand its capabilities and extend the variety of materials and test conditions.

V. HARDWARE DESIGN

Equipment

The Automated Immersion Testing Robotic System primarily consists of a host IBM PC/AT computer interfaced with a Zymark Zymate laboratory robotic work cell. In addition to a robot arm and controller, the robotic work cell is comprised of four major "work stations" designed for performing very specific tasks: (1) a temperature controlled, circulating water bath used to maintain the specimens and immersion liquids at a constant, and often elevated, temperature, (2) a jar capping/uncapping station, (3) a sample blotting station for removing surface droplets from the specimens prior to making weight measurements, and (4) a weighing station for measuring the weight of specimens at timed intervals. A description of each major work station and/or piece of equipment is given in the following paragraphs.

Robot and Controller: The robot and controller module include a Zymate I robot with a "general purpose" hand, a Zymate controller with an expansion memory card and EZLAB programming software, and a teaching pendant for remote positioning.

Robot Grippers: A Zymate general purpose hand was modified with two sets of custom fingers providing a dual capability. General grippers, made from aluminum with rubber padding on the inside, were designed to handle large sample jars from 1-3/4 to 3 inches in diameter. The specific dimensions are illustrated in Figure 2. Forcepslike fingers were designed to handle specimens directly (Figure 3). Also made of aluminum, these fingers are 5 inches in length and can handle a sample thickness of up to 1 inch.

Sample Jars: Standard off-the-shelf 100 mL (2 inches in diameter), Pyrex, screw-cap jars were individually modified to hold the immersion specimens in an upright, vertical position. The jars were modified in two ways: First, a custom, Pyrex specimen holder, consisting of two 1.5-inch concave pieces of glass, was fused 1.25 inches apart with a 2-mm slit opening to the inside bottom of each jar (Figure 4). The specimen holder is fluted at the top, allowing easy replacement of the specimen inside the jar. Second, a Pyrex "tab" was mounted inside the 1.25-inch neck (inside diameter) of the jar, positioned perpendicular to the specimen holder. This tab protrudes out 0.125 inch towards the center of the specimen holder, and is used by the robot to locate the specimen. These modifications allow easy access to the specimens inside the jars, a more uniform flow of the immersion liquid around the specimens, and each specimen to have a unique liquid environment, if desired.

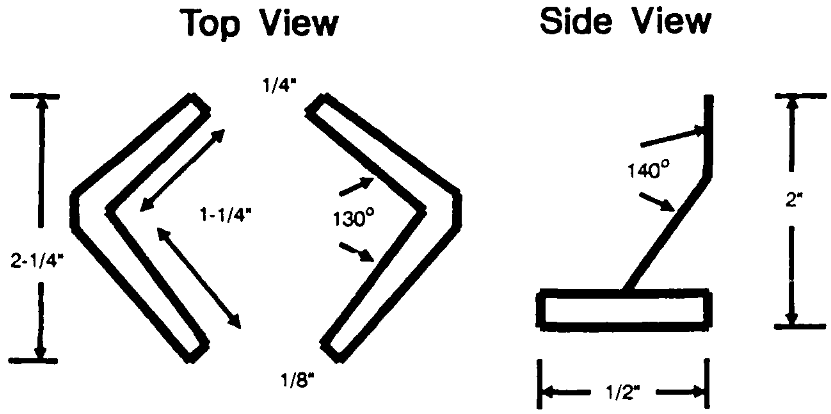

Figure 2. Robot gripper design.

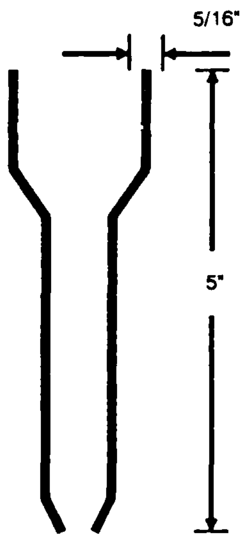

Figure 3. Forceps finger design.

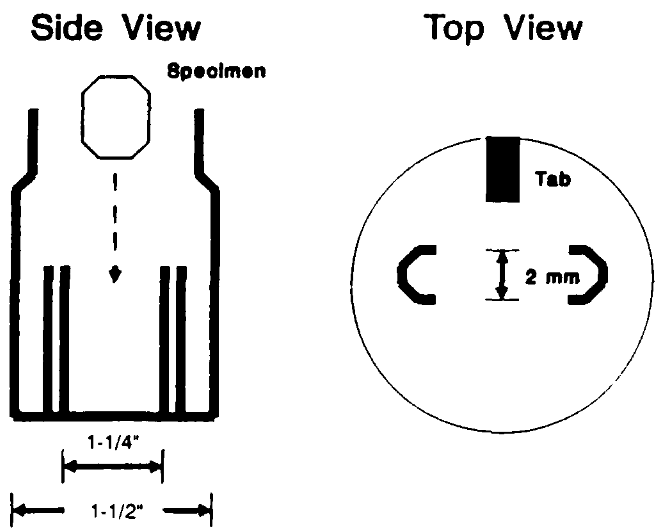

Figure 4. Specimen jar design.

Water Bath: A Fisher Scientific Model 260 temperature-controlled, circulating water bath is used to maintain the desired temperature of the specimens and their immersion liquids. The water bath was modified with a custom-designed rack consisting of an aluminum platform designed to fit within the inner chamber of the bath and a polypropylene platform with uniform jar slots which rests on top of the bath casing. Adjustable cylindrical fittings between the two platforms allow the sample jars to be immersed in the water bath at the appropriate depth.

The specimen jars are placed in the water bath so that only the tops of the jars are exposed. This allows the robot access to the specimen jars and a uniform temperature environment for exposure of specimens.

A blotter is used to remove excess water from the bottoms of sample jars as they are transferred from the water bath. This blotter consists of an absorbent sponge mounted on a polypropylene platform positioned 6 inches above the work bench.

Capping/Uncapping Station: A commercial capping station, Zymark Model Z410, is used for multiple tasks. Conventionally, it is used for capping and uncapping the sample jars. This "work station" is equipped with its own grippers, used to grip the sample jars securely, and a rotational base which rotates in the appropriate direction (clockwise for capping and counterclockwise for uncapping). Using the motion of the capping station, the robot, with a firm grip on the cap, lifts the cap up (or moves the cap downward, whichever the case may be) to complete the capping or uncapping procedures. A polypropylene cap holder, conforming to the shape of the inside of the jar caps, is mounted on a ring stand to provide a secure and accessible place for the robot to store the cap while attending to other tasks.

For our applications, the capping station is also used for "orienting" the jar, and thereby, locating the specimen in the jar. Using the capping station to rotate the jar, the robot's forcepslike fingers are partially inserted into the jar and opened. Contact between the fingers and the glass "tab," located in the neck of the jar, causes the jar to stop rotating and positions the robot fingers at a 90-degree angle directly above the specimen. In this way, the fingers can be lowered and always gain a firm purchase of the specimen for removal and handling.

Blotting Station: The blotting station (Figure 5) simulates the manual method of blotting specimens. The blotter consists of two 3- x 3.5-inch vertically mounted plates. The first plate, held by a ring stand, remains stationary, while the other plate, connected to an air-actuated pneumatic cylinder mounted on a ring stand, is allowed to move back and forth over a distance of 3 inches.

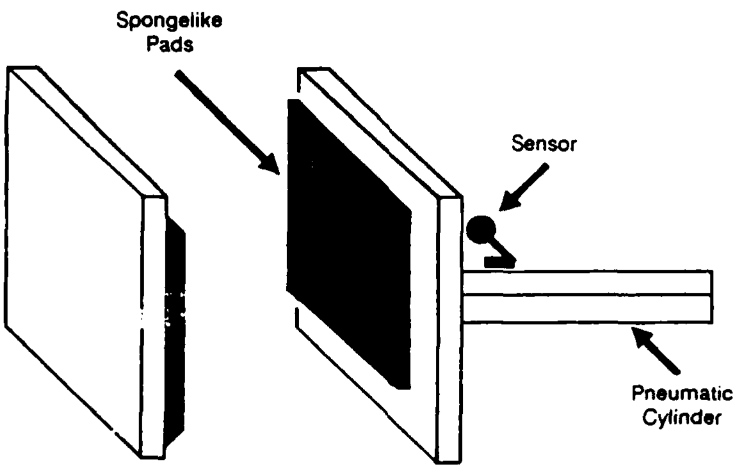

Figure 5. Specimen blotter design.

The design of the blotting station allows the robot's fingers to hold a specimen against a stationary plate while a mobile plate closes in on the specimen. Initially, the plates are positioned 2.75 inches apart. Highly absorbent filter paper is mounted on sponge pads located on the plates' inner surfaces. The filter paper collects surface droplets from the specimen and insures that the specimen edges are blotted as well. A relay switch, mounted to the mobile plate, is used to verify that the blotter is functioning properly at all times.

To insure removal of all surface droplets, each specimen must be regripped and blotted several times. An individual polypropylene specimen holder, mounted 6 inches above the working bench, allows the robot to set the specimen down and regrip it in a different location. This is done so that droplets that may have been trapped beneath the forcepslike fingers are removed. The filter paper, mounted on the mobile blotting plate, is extended 0.25 inch above the mobile plate to allow the robot to blot the inside of the forceps prior to regripping the specimen.

A heated air gun with a directional nose is positioned beneath the blotter to dry the filter paper after each specimen is blotted in preparation for the next specimen.

Weighing Station: A Mettler A163 dual-range, top-loading, analytical balance is used for weighing the specimens. A polypropylene specimen holder is mounted securely to the balance pan and positioned 1 inch below the top of the weighing cavity. Specimens are loaded into the top of the balance via the top sliding window. This window is automatically opened and closed using the Zymark Automatic Balance Door Opener mounted on top of the balance. The automatic door opener, which is connected to the window handle, slides the balance door back and forth using an air-actuated pneumatic cylinder. A relay switch is included for verification purposes.

The Zymark balance interface allows communication between the balance and the robot controller. This allows programmable control for taring the balance prior to each weight measurement.

Sensors: Two optical verification sensors (Model 37622) with LED and photocells are used to verify jar and cap manipulations. A third verification switch is used to verify that the specimens are retrieved from the various "work stations" properly.

Power and Event Controller: This module, Zymark Model Z830, allows programmable control of several logical inputs, switch closures, DC power supplies, and electrical power outlets. It is used for transmitting and receiving information to and from all of the various sensors, relay switches, and solenoid valves used by the Automated Immersion Testing Robotic System.

Host Computer: An IBM PC/AT with a 20 MB hard disk is used for data acquisition and handling. This computer includes a floppy diskette drive, DOS operating system, BASIC programming language, enhanced color monitor and adapter card, IBM Pro-printer, and RS1 graphics package.

The Zymark computer interface card, designed specifically for interfacing the Zymark controller with an IBM PC, allows real-time communication between the robot controller and the host computer.

System Configuration

The layout arrangement is illustrated in Figure 6. The robotic work cell is configured with each work station securely mounted on a 4 x 7 foot tabletop.

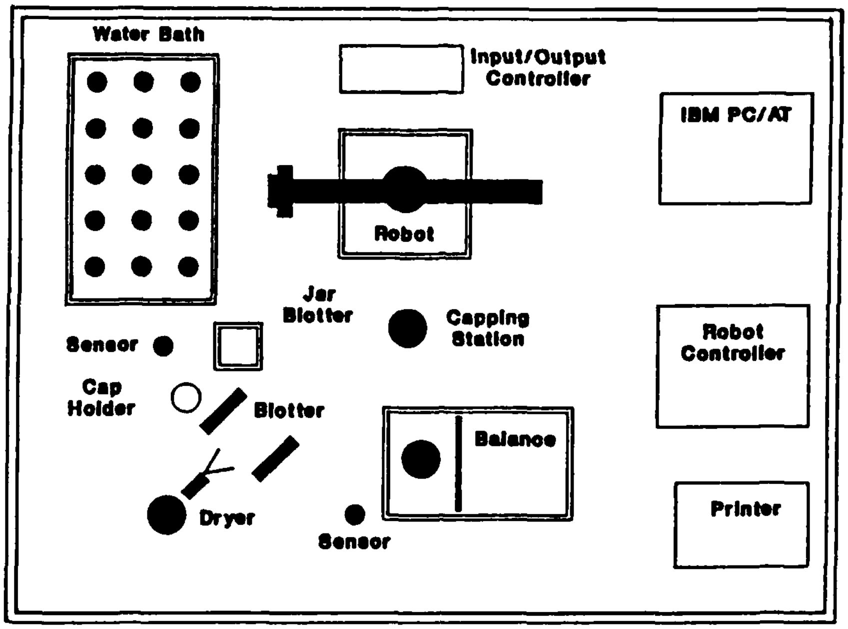

Figure 6. Immersion testing robotic system layout.

Hardware Interfacing/Connections: The robot controller is equipped with 16 communication ports which are necessary for interfacing the individual work stations and various modules into one complete system. Each standard Zymark "work station" and/or module includes an interfacing circuit board. A menu driven programming facility is provided with each interface board. Once these interfaces are connected to a communications port within the robot controller, the necessary programming tools for running and maintaining that particular module are installed. The robot arm, capping station, printer, and power and event controller modules only require a simple connection. The connection is made to the robot controller, via respective interfacing circuit boards and a communications port, to achieve software control over physical functions.

Host Computer: The host computer is interfaced with the robot controller, via a computer interface module connection, to one of the controller's communications ports. Communication between the two systems is achieved with a 9-pin to 25-pin RS232 connection between the controller's computer interface module card and the computer's RS232 communications port.

Pneumatic Cylinders: The blotting and weighing stations are powered by a 100 psi regulated air compressor connected to two 4-way solenoid valves (one for the blotter and one for the balance door opener). Vinyl tubing and a 3-way polypropylene connector connect the valves to the air compressor and the respective pneumatic cylinders. The auxiliary motion of both work stations is programmable by a hardware connection between the solenoid valves and one of the eight switch closures, and the +12-volt power supply on the power and event controller (PEC) module. The status of the verification relay switch mounted on both stations is accessible through a connection between the relay switches and one of the PEC's eight logic inputs. Both of these relay switches are mounted so that the switch is "closed" when the pneumatic cylinders are compressed.

Balance: The communication between the balance (digital output) and the robot controller requires an RS232 interface board connection from the RS232 port of the balance to a communications port in the controller.

Air Dryer: Software control over the air dryer is achieved by plugging in the dryer to a 10-amp electrical outlet which is connected to one of the switch closures and the +5-volt power supply on the PEC.

Sensors: All three optical sensors are connected to a logical input on the PEC. This allows programmable access to the status of each sensor.

The placement of the two primary photo sensors is critical for detecting errors in the robot's routine. Since the majority of problems that may occur with the jar and cap manipulations center around the capping station, both sensors are focused on the capping station at a 90-degree angle to each other. Sensor 1 is positioned at a height required to detect several events: (1) if the cap is not capped properly (i.e., the cap is not threaded properly on the jar), (2) the cap is not removed from the jar during the uncapping procedure (i.e., the cap is fastened too tightly for the robot to release it from the jar), (3) the jar is lifted with the sample while trying to retrieve the specimen from the jar, (4) the jar is retrieved from the bath properly, and (5) the jar is retrieved from the capping station properly.

Sensor 2 is positioned at a height required to detect the following events: (1) the cap is removed from the jar during the uncapping procedure, (2) the cap is removed from the "cap holder" properly in order to recap the jar, (3) the jar is retrieved properly from the bath, and (4) the jar is retrieved properly from the capping station. The specifics of these error detection routines are discussed in greater detail in Section VI, Software Design.

VI. SOFTWARE DESIGN

The software design of the Automated Immersion Testing Robotic System is made up of two primary modules, the Method Controller and the Manager. The Method Controller is a set of subroutines written in the EZLAB programming language in the robot controller and used to instruct the robot through the mechanics of the immersion testing procedure. The Manager, written in BASIC on the IBM PC/AT, provides control of the sequence and scheduling of events, and handles all data acquisition. The Manager is the host program which calls on the Method Controller to instruct the robot through the appropriate set of tasks.

Method Controller

The main function of the Method Controller is to instruct the robot through the immersion testing procedure and handle the mechanics of the error detection, verification, and correction routines. All discrete robot movements are assigned and programmed using the robot controller. Robot functions, or tasks, are programmed as subroutines using the EZLAB programming language. The organization of these tasks, which together make up the immersion test procedure, are coordinated in the Manager using BASIC on the IBM PC/AT.

A unique subroutine is created for each discrete task the robot performs. The primary subroutines which make up the immersion testing procedure of an immersion study are as follows:

1. remove sample jar from water bath,

2. blot bottom of jar (remove excess water from the bottom of the jar),

3. check for presence of jar (using two photo sensors, four positions on the jar are verified to ensure that the robot is handling the jar properly),

4. uncap the jar,

5. check for successful removal of cap,

6. place cap aside,

7. remove sample from jar,

8. check for successful removal of sample,

9. blot sample (remove surface droplets),

10. check for sample (after blotting procedure),

11. weigh sample (recording the weight),

12. check for sample removal (from balance),

13. place sample in jar,

14. cap sample jar,

15. check for presence of jar, and

16. return sample jar in bath.

Since the Manager makes the necessary decisions when procedural errors occur in the immersion test method routine, the above 16 primary subroutines are combined in higher level programs in the robot controller so that the last task in each program is a verification, or error checking, routine. When the Method Controller returns control to the Manager, the status of the sensor used for verification is also returned. This allows the Manager to make the appropriate decision and instruct the Method Controller to execute the most suitable action or correction subroutine. All subroutines and programs used to instruct the robot through the mechanics of the immersion test method are listed in Appendix A.

Manager

The Manager handles the scheduling activities, problem/error decisions, calculations, data storage and retrieval, and reporting features. The Manager, Appendix B, is divided into four primary sections: (1) the Initialization Procedure, (2) the Main Module or Scheduling Procedure, (3) the Immersion Testing Module, and (4) Function Key Subroutines. The Manager continuously searches for the next activity to be performed until one of the function keys, set up to terminate the program, is executed.

Initialization: The initialization portion of the program assigns all variables and flags, opens communication between the IBM and robot controller, opens all data files used to keep track of specimens, and initializes function keys (see section on function key subroutines). Six types of data files are used to keep track of specimens, events, and errors.

STUDIES.DAT is the data file which keeps track of all current specimens undergoing an immersion test. Each record in the data file contains information unique to a particular specimen: (1) the location (index) within the water bath; i.e., which jar contains the specimen, (2) the initial weight of the specimen, (3) the time the last weight measurement was made, (4) the date the last weight measurement was made, (5) the current time interval between weight measurements, (6) the time the next weight measurement should be made, and (7) the date the next weight measurement should be made.

JARSTATS.DAT is the data file which indicates the current status of each jar in the water bath (in this case, there are 30 bath locations indexed in a uniformed 6 x 5 rack). Each record (the record numbers correspond to the index of the jars within the bath) in this data file contains information unique to each specimen jar: (1) the rating of the jar (Since some jars are more reliable than others, they are rated so that specimens requiring more frequent weight measurements use the most reliable jars, and specimens weighed less often use less reliable jars. This limits the number of errors that occur while removing and replacing specimens in their immersion jars. A rating of "1" indicates the most reliable jars to be used for specimens which require measurements every 24 hours or less. A rating of "2" indicates the less reliable jars to be used for specimens which have a time interval between weighings of 25 or more hours.), (2) a flag to indicate if the jar is currently being used (1 indicates yes, 0 indicates no), (3) the current time interval between weighings assigned to the specimen in that particular jar, and (4) whether or not the jar contains the immersion liquid [1 indicates yes, 0 indicates no (empty)].

STATUS.REP is the data file which is for diagnostic purposes. As errors occur, a record of (1) the type of error, [0 indicates nonfatal, 1 indicates fatal (i.e., the study was discontinued for a particular specimen)], (2) which specimen/jar had the error, (3) the time of the error, and (4) a description of the error (what steps were taken to correct the error and if the error correction routines were successful). This data file is used by the operator in order to locate and fix, if appropriate, problems so that a study may be continued, and to locate problem jars (i.e., change their status) or problems in the routines which may require modifications.

DATAFILE.DAT is the data file which keeps track of file names used to store the raw data for each specimen. Each record of the data file corresponds to the unique specimen which occupies that record number (index) in the water bath. The name of the data files are assigned by the operator when a specimen is initialized.

The data files (unique for each specimen) contain the raw data for a unique specimen. Each record contains information related to one weight measurement: (1) the date of the measurement, (2) the time of the measurement, (3) the number of hours in the study past, (4) the weight measurement taken, and (5) the change in weight (current weight - initial weight).

SPECS.DAT is the data file which stores all information concerning the specimens currently undergoing the immersion test. Each record contains information unique to each specimen (the record number corresponds to the specimen/jar index number): (1) description of specimen, (2) type of resin, (3) type of fiber, (4) composite prepreg configuration (i.e., direction of fibers), (5) temperature of study, (6) specimen thickness, (7) specimen surface area, and (8) specimen conditioning parameters.

Main Module: The main module makes use of the STUDIES.DAT file by looping through all of the file's records. A comparison of the current date and time is made with the "next date" and "next time" fields of each record. If there is a match, or the current time and date is greater than the time and date of the next scheduled weighing, the immersion testing subroutine is invoked. After one pass through all records of the file (invoking the immersion testing subroutine as appropriate), this module checks to see if the operator has initiated any of the function keys (i.e., the operator, through the function keys, takes priority over the automation of the program).

There are seven function keys set up for the operator to: (1) stop the program and release control of the robot controller, (2) stop the program without releasing the controller, (3) plot/report the data on a particular specimen gathered so far, (4) start a new specimen, (5) initialize/change the status of the specimen jars in JARSTATS.DAT, or (6) print errors encountered thus far.

If no specimen at the current time needs a weight measurement, the robot is instructed to fill empty specimen jars with the immersion liquid (i.e., immersion liquids must reach the testing temperature before a specimen is initialized into the immersion study). The flow of this module is illustrated in the flow diagram in Figure 7.

The operator makes the decision to terminate the study on a particular specimen. In the event of termination, the operator terminates the Manager module, removes the specimen, uses function key 5 to change the status of that particular specimen jar, and restarts the Manager module. Once the jar's status has been altered appropriately, the system is designed to recognize the change and reuse the jar as needed.

Immersion Testing Subroutine: The immersion testing subroutine is separated into several categories or sets of lower level subroutines: (1) the primary subroutine for initiating a weight measurement on a specimen and all data acquisition, (2) the controller subroutines for communicating with the robot controller, setting variables, retrieving values of variables, and invoking EZLAB robot subroutines, (3) control routines for sequencing events for performing the immersion weight measurements (i.e., instructing the robot through the appropriate seq-uence of events and recognizing errors when they occur), and (4) error handling routines which are executed if errors are detected in the robot's procedure.

The abstract algorithm for making a weight measurement is as follows:

- The bath location (index) is set appropriately in the robot controller.
- The program to take a weight measurement is executed.
- The weight of the specimen is retrieved from the robot controller.
- The file name for data storage is retrieved from DATAFILE.DAT.
- The data file is opened and the current date, time, weight, calculated total number of hours in study, and calculated change in weight values are stored in the data file.
- The next weighing time and date are calculated and the time interval between weight measurements is increased, if appropriate. Currently, the algorithm for increasing the time interval takes the last five weight measurements of the study and averages them. If all five weight measurements are within + 0.002 g of the mean, then the time interval between weight measurements is increased 100%. This information is then updated in STUDIES.DAT in the record associated with the specimen.
- Return to main program.

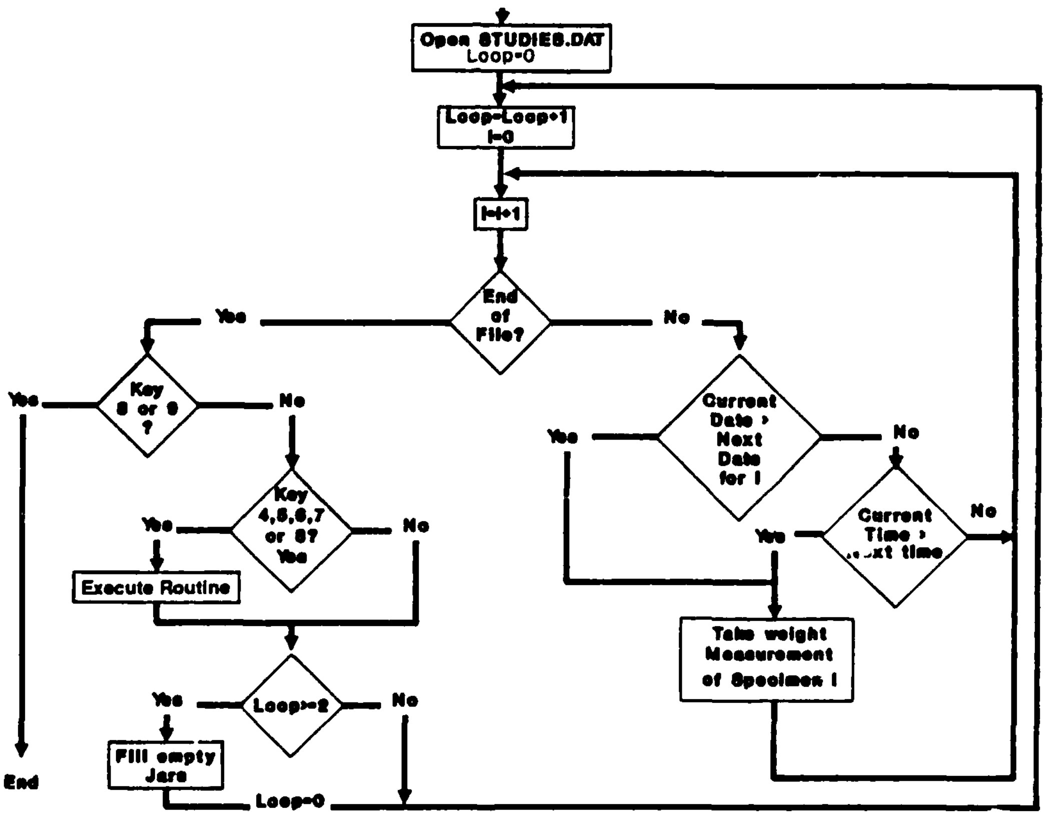

Figure 7. Immersion scheduling flow diagram.

Controller and Control Subroutines: The controller subroutines consist of repetitive portions of code primarily used for sending and retrieving information to and from the robot controller. The control subroutines for running the immersion test are the actual system calls used to instruct the robot through the test procedure. This section of the code also includes all the error handling and decision sequencing.

Function Key Subroutines: Function keys 9 and 10 are used by the operator to terminate the Manager. If the operator needs to make changes in the Method Controller or reprogram robot positions, function key 10 must be used to release the control that the IBM has over the robot controller before the controller's keyboard will accept information. Function key 9 may be used when the operator does not require access to the robot controller.

Function keys 4, 5, and 6 perform their specific functions and return to the Manager's control. Function key 4 dumps the text found in STATUS.REP to the line printer and deletes the file. All information relating to errors is printed on the computer's line printer: (1) which jars or specimens encountered the errors, (2) what steps were taken to correct the problem, and (3) if the correction routines were successful. Function keys 5 and 6 are used to initialize and change, respectively, the flags which are used by the system to keep track of specimen jars.

The two most frequently used function key routines are function key 7 (starting a new specimen) and function key 8 (plotting data). The flow diagram, Figure 8, illustrates the algorithm for initializing a new specimen. Since some of the custom specimen jars are more reliable than others, jars are rated according to their reliability. In the process of initializing a new specimen to the system, each jar is evaluated sequentially in the following manner:

- If rating = 2 (less reliable), then next jar.
- If the jar is not in use and contains immersion liquid, then proceed with that jar.
- If the jar is not in use but does not contain the immersion liquid, then remember as a possible jar, but continue to next jar.
- If the jar is in use but time intervals between weighings is less than 24 hours, then remember as a possible jar for SWITCHING, but continue to next jar.

The most desirable situation is that a jar of a reliable rating is not being used and contains the immersion liquid. If such a jar is not available, then the next situation most desirable is a jar with a reliable rating which is empty, in which case the robot fills the jar, sets the timer for 1 hour to allow the immersion liquid to reach the desired temperature (i.e., that of the water bath), and moves on to other duties. When the hour is ended, the robot signals the operator and the initialization procedure continues.

If neither of the above two situations is true for any jar, then a jar which is currently used with a time interval greater than 24 hours is investigated. First, the system searches the status flags of each jar in JARSTATS.DAT looking for an unused jar with a rating equal to 2 (less reliable). If such a jar is found, the robot retrieves the first jar from the water bath (with a rating of 1 where time intervals are greater than 24 hours), removes the specimen, empties the jar, fills the jar with the immersion liquid, returns the jar to the water bath, and sets the timer for 1 hour. Next, the robot retrieves the second jar (with a rating of 2, and containing the immersion liquid), places the specimen in the jar, and returns the jar to the water bath. Again, the robot will move on to other tasks, as appropriate, until the timer indicates that 1 hour has passed. The robot signals the operator and the initialization procedure continues.

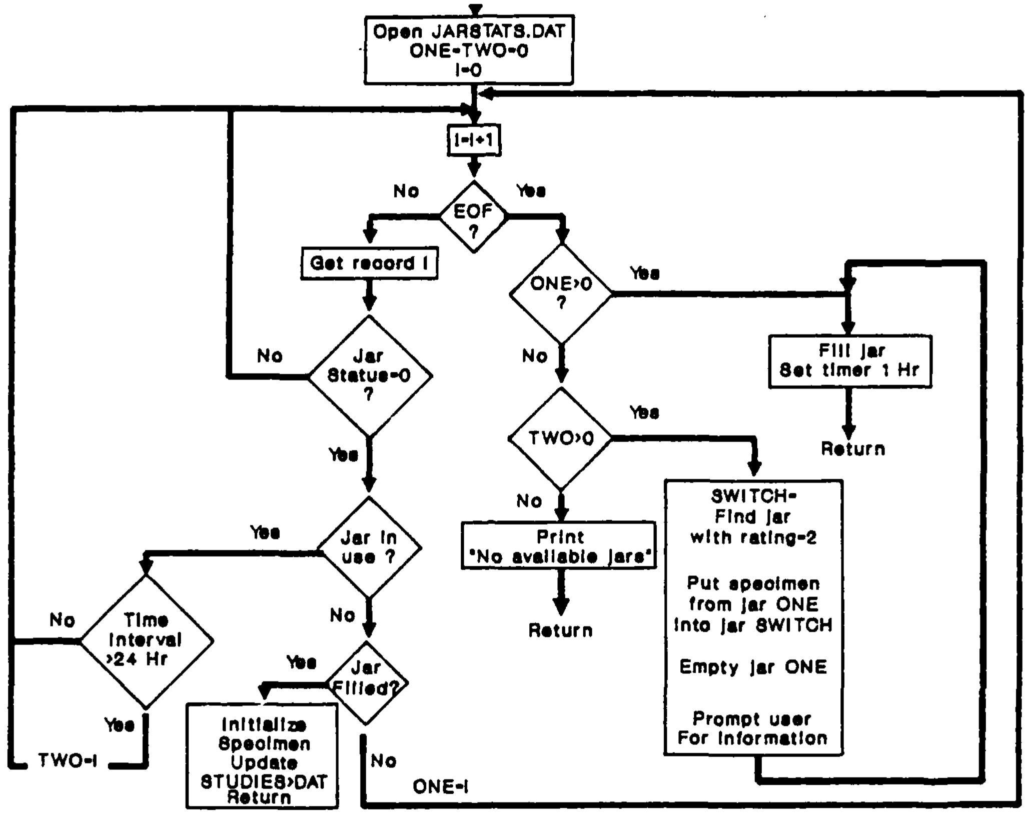

Figure 8. Initializing immersion study flow diagram.

Once the robot locates a jar for the new specimen, the system prompts the user for initial information and makes the appropriate changes in the files STUDIES.DAT, DATAFILE.DAT, JARSTATS.DAT, and SPECS.DAT.

The operator may use function key 8, at any time, to plot the data collected thus far for any specimen currently registered in the system. Function key 8 reformats the necessary information found in the specimen's unique data file, and uses the RS1 plotting language to plot the change in weight versus the square root of time (Figure 9 in Section VII, System Verification). Control is returned to the Manager at the operator's request.

Comments: The operator has the ability, through the use of the function keys, to override the Manager's control over the scheduling of events. Calculated times for the next weight measurements may not be precisely met; however, the times recorded for each weight measurement taken are the actual time of the measurements and not the calculated times.

VII. SYSTEM VERIFICATION

Testing and Verification

Two studies were performed to verify and test the reliability of the Automated Immersion Testing Robotic System. The first study involved performing accelerated immersion tests (the immersion liquid was kept at a constant temperature of 60°C) on four glass-reinforced composite specimens machined from the same sheet of material. Two specimens were tested on the immersion system, while the other two specimens were tested by a human operator. This was done to compare the accuracy and to determine the robot's ability to duplicate, if not improve, the blotting and weighing routines. Weight measurements on the specimens were taken at the same time throughout a 3-month period, but not overnight or on weekends. The same water bath and sample jar types were used, and manual weight measurements were made with the same model of Mettler balance. Figures 9a and 9b show graphical results of specimens tested using the automated and manual methods, respectively. A comparison of individual data points indicates that the automated system can perform immersion studies with the same, if not better, accuracy as obtained by the manual method. Reliability was demonstrated in that the robotics system functioned autonomously during the test without committing an uncorrectable error. It is also apparent that the automated blotting procedure was performed reliably, since a single droplet of water left on the specimen during weighing produces a well-defined "outlier" data point and can be interpreted as such by the computer during analysis.

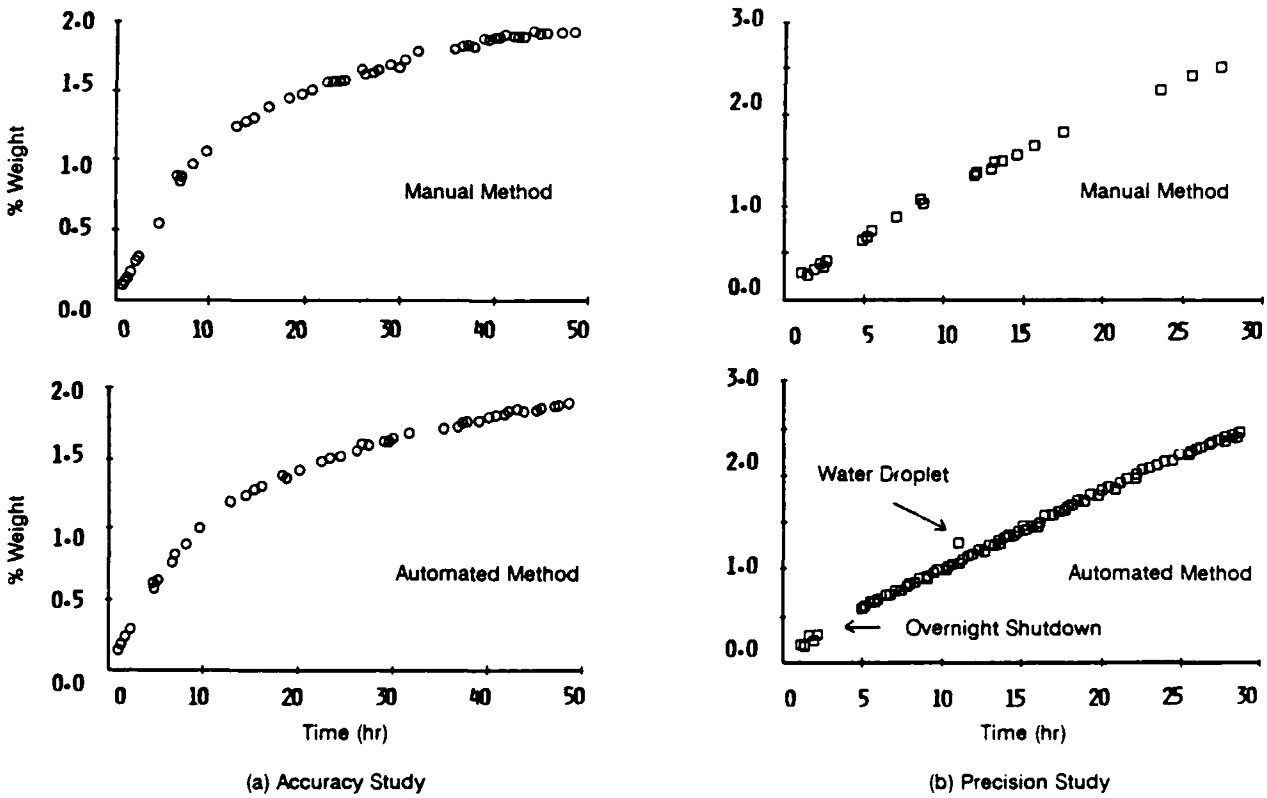

Figure 9. Immersion test data of an epoxy resin/aramid fiber laminate.

The second study involved performing accelerated immersion tests on four aramid-reinforced composite specimens. All four specimens were dried in a desiccator prior to the test. The only difference in this study was that the robot was allowed to run overnight and on weekends, while the operator kept to the same schedule as in the previous study. By increasing the number of weight measurements and making measurements during evenings and weekends, the Automated Immersion Testing Robotic System was able to demonstrate an improvement in the precision and reliability of test results compared to the manual method as shown in Figure 9b.

Productivity Enhancement

A computer simulation was performed to illustrate the increase in productivity of the automated test method over the manual test method. The parameters used for each simulation are listed in Table 1. One of the advantages of the automated test method for performing immersion studies is that more weight measurements may be taken in the course of a single study. However, in order to compare the productivity of the two test methods, several constraints and assumptions were made. Since an immersion test takes the same number of days to complete a single sample independent of the test method, the automated method was constrained to the same number of weight measurements per specimen as the manual method. For this particular simulation, 15 weight measurements were taken in a period of 101 days for each specimen. It was assumed with the manual method that an operator works 8 hours per day and 47 weeks per year (i.e., 2 weeks of holidays, 2 weeks of vacation, and 1 week of sick leave) performing nothing but immersion testing tasks. For the manual method, it was also assumed that data handling (i.e., data entry and plotting/evaluating on a computer) is performed only after the 101-day study is completed. An operator requires 0.167 minute to enter the set of time and weight measurements in the computer, and 60 minutes to plot and evaluate each specimen. The automated method was allowed to perform 52 weeks per year, assuming no downtime, and the data handling time was assumed to be negligible. No constraints were put on the number of specimens that the automated system or manual system could handle at any time (i.e., it was assumed that there was an infinite number of specimen jars and bath slots). The manual blotting and weighing procedure takes approximately 2.5 minutes per specimen, while the automated blotting and weighing procedure takes approximately 6.0 minutes per specimen.

Table 1. SIMULATION PARAMETERS

Parameter	Manual Method	Automated Method
Time/Weight Measurement (min)	2.5	6.0
Working Hours/Day	8.0	24.0
Working Days/Week	5.0	7.0
Working Weeks/Year	47.0	52.0
Data Acquisition:		
Time/Specimen Plotted (min)	60.0	0.0
Data Entry (min)	0.167 * No. of Points	0.0

The simulation for each test method was performed to quantify the number of specimens that could be studied completely in one year. On the average, a human operator makes 29 weight measurements in an 8-hour day, while the Automated Immersion Testing Robotic System handles, on the average, 237 weight measurements in a 24-hour day. Thus, on a yearly basis, only 266 specimens can be tested manually while the automated system can test 2,775 specimens. Through automation, productivity has been increased by 1,170%.

VIII. FUTURE ENHANCEMENTS

Currently, plans for extending the Automated Immersion Testing Robotic System are underway. Figure 10 illustrates the design of an integrated system for materials durability testing. This system will consist of four robot arms working simultaneously with two of the robots performing immersion testing as described in this report. Plans include the following additional capabilities: (1) automated specimen conditioning using a vacuum oven, (2) automated recycling of specimens, (3) a freeze-thaw study capability, (4) automated thickness measurements, (5) dry heat studies, (6) nondestructive testing, and (7) a variety of immersion temperature conditions. Enhanced software capabilities shall include: (1) sophisticated data analysis, (2) enhanced decision making capabilities through expert system development, and (3) machine learning capabilities.

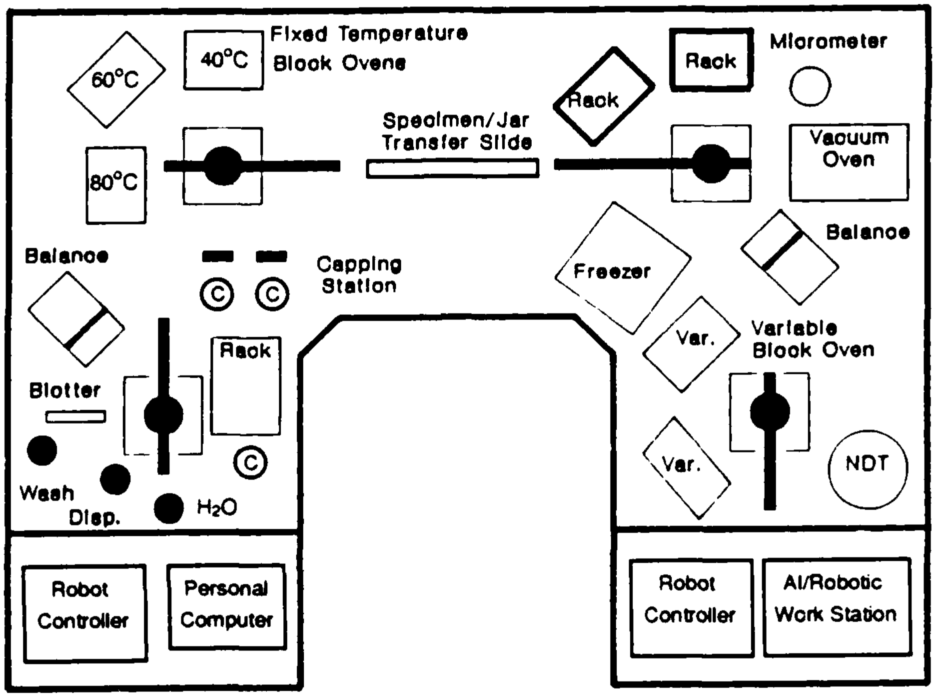

Figure 10. Durability testing robotic system layout.

Robot A shall consist of: (1) a rack for holding unused specimen jars, (2) a station for immersion liquid disposal and liquid renewal, (3) a wash station for washing immersion specimens prior to weighings (e.g., if immersion liquid is a brine solution), (4) a specimen blotter, (5) a capping station, and (6) an analytical balance. Robot B shall consist of three fixed temperature block ovens set at 40°C, 60°C, and 80°C for maintaining immersion liquid temperatures and two capping stations.

To increase the number of weight measurements that can be performed in a 24-hour period, the immersion test procedure will be performed by dividing the test method tasks between robots A and B. Robot B shall be in charge of specimen jar manipulation into, and out of, the block ovens, and removing and replacing specimens in their respective containers. Robot A, once receiving a specimen from robot B, shall be in charge of blotting, washing, and weighing procedures. In addition to the immersion testing tasks, robot A shall also be in charge of jar preparations (i.e., cleaning, filling, and disposing of immersion liquids).

Specimens designated for accelerated aging studies shall be obtained from weathering chambers and environmental test sites, and will be placed on racks and introduced into the Durability Testing System at location C. Robot C shall be in charge of performing any further specimen-conditioning procedures, taking initial weight measurements, measuring specimen thicknesses, and handling specimen recycling/freeze-thaw procedures.

Robot D shall consist of variable temperature block ovens for performing dry-heat studies, and an NDT system for obtaining qualitative information on the moisture content and deterioration (delamination, crack formation, etc.) of immersion specimens. Several NDT methods are under investigation. Location D has been designed to allow system expansion. In the future, additional information may be obtained by incorporating dynamic mechanical analysis and chromatographic techniques for examining immersion specimens and liquids, respectively.

The Durability Testing System will allow numerous variations in specimen treatment/testing conditions. Specimens undergoing immersion testing may be dried and recycled, desorption measurements can be made, or the effects of freezing or high temperature treatment on NDT and sorption-diffusion properties may be studied. Complete records will be compiled for each specimen and a database will summarize results obtained for other materials/specimens. Depending upon relative changes in measurements and anticipated behavior, the robotics system may modify a specimen's test plan or alert the user of unusual behavior. When completed, the durability system will be integrated with a multitasking, multiprocessor, AI/Robotic work station.

REMOVE.JAR

```
      get.jar.from.bath
      blot.jar
      check.for.jar
```

GET.JAR.FROM.BATH

```
      gp.hand
      move.to.bath.no.jar
      over.cap.in.bath
      grip.cap
      raise.jar.from.bath
      if bath.location<13 then 50
      move.to.clearing
50
```

BLOT.JAR

```
      move.to.blot.jar
      lower.jar.to.sponge
      set timer 1 3 seconds
      wait for timer 1
      move.to.blot.jar.2
```

CHECK.FOR.JAR

```
      slant=0
      jar.flag=0
      sensor2.front
      if photo.sensor.2-0 then 50
      jar.flag=jar.flag+1
      slant=1
 50   sensor2.back
      if photo.sensor.2-0 then 100
      jar.flag=jar.flag+1
      slant=2
100   move.to.sensor1
      sensor1.front
      if photo.sensor=0 then 150
      jar.flag=jar.flag+1
      slant=3
150   sensor1.back
      if photo.sensor=0 then 200
      jar.flag=jar.flag+1
      slant=4
200   if slant=0 then 250
250
```

UNCAP.JAR

```
      no module wait
      move.to.capping.jar
      module wait
      open.capping.grippers
      gp.hand
      close.capping.grippers
      repos.grippers.around.cat
      grip.cap
      uncap
```

UNCAP

```
      no module wait
      turn.to.uncap
      module wait
      robot.up.6mm
      robot.up.8mm
      robot.up.10mm
      robot.up.12mm
      lift.cap
```

CAP.ASIDE

```
      avoid.sensor
      place.cap.in.holding
```

PLACE.CAP.IN.HOLDING

```
      move.to.cap.holding
      lower.cap
```

GET.SAMPLE.FROM.JAR

```
move.to.capping.forceps
forceps.hand
orient.jar
ready.forceps.for.sample
close.capping.grippers
set timer 1 2 seconds
wait for timer 1
open.capping.grippers
lower.forceps.to.sample
grip.sample
set timer 1 2 seconds
wait for timer 1
lift.s
set timer 1 2 seconds
wait for timer 1
lift.sample.out.of.jar
```

CHECK.FOR.SAMPLE.FROM.JAR

```
   sample.flag=0
   clear.beam.sensor
   move.to.sensor.sample
   set timer 1 2 seconds
   wait for timer 1
   if beam.sensor=0 then 30
   sample.flag=1
30 clear.beam.sensor
```

BLOT.FORCEPS.TIPS

```
forceps.hand
move.to.jar.blot.forceps
blot.tips
set timer 1 2 seconds
wait for timer 1
move.to.jar.blot.forceps
```

BLOT.INSIDE.FORCEPS

```
forceps.hand
```

```
gp.hand
move.to.cap.holding
```

ORIENT.JAR

```
lower.forceps
open.capping.grippers
open.forceps.orient
turn.to.orient
```

GET.SPECIMEN

```
raise.arm.above.holding
forceps.hand
move.to.capping.forceps
orient.jar
forceps.hand
move.to.capping.forceps
dryer.on
blot.forceps.tips
move.towards.blotter
blot.forceps
dryer.off
to.sample.holder
get.sample.from.holder.1
check.for.sample.from.blotter
```

BLOT.FORCEPS

```
forceps.hand
blot.inside.forceps
move.in.blotter
set timer 1 10 seconds
wait for timer 1
move.towards.blotter
```

GET.SAMPLE.FROM.HOLDER.1

```
sample.in.holder
```

```
    move.to.sample.blotter
    if blot.f.flag=1 then 5
    move.to.left.blotter
    move.down.to.blotter
    goto 10
 5  move.to.left.blotter.2
    move.down.to.blotter.2
10  close.forceps
    set timer 1 1 seconds
    wait for timer 1
    forceps.hand
    if blot.f.flag=1 then 15
    move.to.left.blotter
    goto 20
15  move.to.left.blotter.2
20  move.to.sample.blotter
```

BLOT.SAMPLE

```
    blot.f.flag=0
    blot.flag=0
    error.blot=0
    blot.sample.center
    if error.blot>0 then 100
    put.sample.in.holder.1
    blot.inside.forceps
    get.sample.from.holder.2
    blot.sample.left
    if error.blot>0 then 100
    put.sample.in.holder.2
    blot.f.flag=1
    blot.inside.forceps
    blot.f.flag=0
    get.sample.from.holder.3
    blot.sample.right
    if error.blot>0 then 100
    put.sample.in.holder.3
    get.sample.from.holder.1
100
```

OPEN.BLOTTER

```
    error.blot=0
    count=0
10  count=count+1
    if count>5 then 50
    blot.open
    if blotter.switch=1 then 10
    goto 100
50  error.blot=1
100
```

```
    grip.sample
    to.sample.holder.out
```

CHECK.FOR.SAMPLE.FROM.BLOTTER

```
    sample.flag=0
    move.to.sensor.sample
    set timer 1 2 seconds
    wait for timer 1
    if beam.sensor=0 then 30
    sample.flag=1
30  clear.beam.sensor
```

BLOT.SAMPLE.CENTER

```
    hold.coount=1
    move.to.sample.blotter
    lower.sample.in.blotter
    module wait
    move.sample.right
    blot.flag=1
    close.blotter
    if error.blot=1 then 50
    set timer 1 1 seconds
    wait for timer 1
    blot.flag=2
    open.blotter
    if error.blot=1 then 50
    lower.sample.in.blotter
50
```

CLOSE.BLOTTER

```
    error.blot=0
    count=0
 5  count=count+1
    if count>5 then 50
    blot.close
    if blotter.switch=0 then 5
    goto 100
50  error.blot=1
100
```

PUT.SAMPLE.IN.HOLDER.1

```
to.sample.holder
sample.in.holder
forceps.hand
to.sample.holder
```

PUT.SAMPLE.IN.HOLDER.2

```
to.sample.holder.2
sample.in.holder.2
forceps.hand
to.sample.holder.2
```

PUT.SAMPLE.FROM.HOLDER.3

```
to.sample.in.holder.3
sample.in.holder.3
forceps.hand
to.sample.holder.3
```

BLOT.SAMPLE.RIGHT

```
    hold.count=3
    to.blotter.outside
    outside.right
    blot.flag=1
    close.blotter
    if error.blot=1 then 50
    set timer 1 1 seconds
    wait for timer 1
    blot.flag=2
    open.blotter
    if error.blot=1 then 50
    to.blotter.outside
50
```

WEIGH.SAMPLE

```
if weight.type=1 then 1
move.to.balance.out
```

GET.SAMPLE.FROM.HOLDER.2

```
to.sample.holder.2
sample.in.holder.2
grip.sample
to.sample.holder.2
```

GET.SAMPLE.FROM.HOLDER.3

```
to.sample.holder.3
sample.in.holder.3
grip.sample
to.sample.holder.3
```

BLOT.SAMPLE.LEFT

```
    hold.count=2
    to.blotter.inside
    inside.right
    blot.flag=1
    close.blotter
    if error.blot=1 then 50
    set timer 1 1 seconds
    wait for timer 1
    blot.flag=2
    open.blotter
    if error.blot=1 then 50
    to.blotter.inside
50
```

TAKE.WEIGHT

```
tare.balance
weigh.sample
```

OPEN.BALANCE

```
  count=0
5 count=count+1
```

```
    goto 2
1   move.to.balance
2   error.balance=0
    door.open.flag=1
    door.close.flag=0
    open.balance
    if error.balance>0 then 100
    if weight.type=1 then 3
    sample.in.balance.out
    goto 4
3   sample.in.balance
4   forceps.hand
7   move.to.balance.out
8   door.close.flag=1
    close.balance
    if error.balance>0 then 100
9   set timer 1 10 seconds
    wait for timer 1
    if wieght.type=0 then 10
6   mw.sample=weight.sample
    goto 20
10  iw.sample(bath.location)=
    weight.sample
    if iw.sample(bath.location)=
    -999 then 20
20  door.open.flag=2
    open.balance
    if error.balance>0 then 100
    sample.in.balance.out
    grip.sample
    move.to.balance.out
    door.closed.flag=2
30  close.balance
100
```

```
    if count>5 then 50
    door.open
    if door.status=1 then 5
    goto 100
50  if door.open.flag=2 then 66
    error.balance=1
    goto 100
66  error.balance=2
100
```

CLOSE.BALANCE

```
    count=0
5   count=count+1
    if count>5 then 50
    door.close
    if door.status=0 then 5
    goto 100
50  if door.closed.glag=2 then 75
    error.balance=3
    goto 100
75  error.balance=4
100
```

CHECK.FOR.SAMPLE.FROM.BALANCE

```
    sample.flag=0

    clear.beam.sensor
    move.to.sensor.sample
    set timer 1 2 seconds
    wait for timer 1
    if beam.sensor=0 then 50
    temp.weight=weight.sample
    if temp.weight<0.1 then 45
    pump.on
    open.balance
    move.to.balance
    forceps.hand
    sample.in.balance

    grip.sample
```

PUT.SAMPLE.IN.JAR

```
    lift.sample.out.of.jar
    lift.halfway
    open.forceps.slightly
    move.to.capping.forceps
```

GET.CAP

```
    dryer.on
    blot.forceps.tips
    get.cap.from.holding
    avoid.sensor
    move.to.capping.cap
```

CAP.JAR

```
     move.to.balance
     pump.off
     move.to.sensor.sample
     set timer 1 2 seconds

     wait for timer 1
     if beam.sensor=0 then 50
45   sample.flag=1
     stop.immersion.jar(bath.location)=1
50   clear.beam.sensor
```

CAP.SAMPLE.JAR

```
close.capping.grippers
move.to.capping.cap
place.cap.on.jar
no module wait
turn.capper.cap
module wait
robot.down.4mm
robot.down.6mm
robot.down.8mm
robot.down10mm
robot.down.12mm
set timer 1 3 seconds
wait for timer 1
```

CHECK.FOR.JAR.2

```
     slant=0
     jar.flag=0
     sensor1.fron
     if photo.sensor=0 then 50
     jar.flag=jar.flag+1
     slant=3
50   sensor1.back
     if photo.sensor=0 then 100
     jar.flag=jar.flag+1
     slant=4
100  move.to.sensor1
     sensor2.back
     if photo.sensor.2=0 then 150
     jar.flag=jar.flag+1
     slant=2
150  sensor2.front
     if photo.sensor.2=0 then 200
     jar.flag=jar.flag+1
     slant=1
200  move.to.blot.jar
```

```
cap.test=0
cap.sample.jar
```

GET.CAP.FROM.HOLDING

```
gp.hand
move.to.cap.holding
lower.cap.2
grip.cap
move.to.cap.holding
```

GET.JAR

```
open.capping.grippers
move.to.capping.jar
check.for.jar.2
```

REPLACE.JAR

```
move.to.blot.jar
dryer.off
put.sample.jar.in.bath
clear.all
gp.hand
```

PUT.SAMPLE.JAR.IN.BATH

```
    if bath.location<13 then 10
    move.to.clearing
10  move.to.bath.jar
    lower.jar.in.bath
    gp.hand
    move.to.bath.no.jar
```

REGRIPPING

```
clear.sensor.2
regrip.jar
move.to.blot.jar
check.for.jar
```

PUT.JAR.IN.BATH

```
move.to.blot.jar
dryer.off
put.sample.jar.in.bath
clear.all
gp.hand
```

ERROR.BLOT.ROUTINE

```
lower.sample.in.blotter
move.to.sample.blotter
```

REGRIP.JAR

```
clear.all
box1
box2
gp.hand
box3
close.gp.hand
box4
box5
box6
box7
box8
box9
box10
box11
box12
box11
box4
box13
gp.hand
box14
grip.cap
box15
clear.all
```

JAR.HANG.2

```
lift.halfway
grip.sample
close.capping.grippers
move.to.capping.forceps
```

NEXT.CAP

```
get.extra.cap
avoid.sensor
place.cap.in.holding
get.cap.from.holding
avoid.sensor
move.to.capping.cap
```

GET.EXTRA.CAP

```
e.cap1
e.cap2
gp.wide
e.cap3
e.cap4
grip.cap
e.cap3
e.cap2
e.cap1
```

UNCAPPED.JAR.ASIDE

```
get.jar.from.capping.no.cap
clear.blot
put.jar.aside.no.cap
```

GET.JAR.FROM.CAPPING.NO.CAP

```
gp.hand
move.to.capping.to.grip.jar
move.to.grip.jar
open.capping.grippers
grip.jar
lift.jar.no.cap
```

```
PUT.JAR.ASIDE.NO.CAP

        regrip1
        regrip3
        gp.hand
        regrip1
```

```
CAP.ASIDE

        avoid.sensor
        place.cap.in.holding
```

```
FILL.JAR

        w.pump.on
        move.to.dispenser
        valve.on
        set timer 1 30 seconds
        wait for timer 1
        valve.off
        w.pump.off
        move.away.from.dispenser
```

```
MOVE.TO.DISPENSER

        over.to.dispenser
        in.to.dispenser
        up.to.dispenser
```

```
MOVE.AWAY.FROM.DISPENSER

        in.to.dispenser
        over.to.dispenser
```

```
PUT.JAR.IN.CAPPING.NO.CAP

        move.to.capping.to.grip.jar
        open.capping.grippers
        move.to.grip.jar
        gp.hand
        lift.cap
        close.capping.grippers
```

```
SET.SAMPLE.ASIDE

        clear.beam.sensor
        blot.sample.center
        put.sample.in.holder.1
        gp.hand
```

```
EMPTY.JAR

        move.to.beaker
        mb.2
        mb.3
        mb.4
        mb.5
        mb.6
        set timer 1 3 seconds
        wait for timer 1
        mb.4
        mb.3
        mb.2
        move.to.beaker
        gp.hand.jar
```

```
80      ;Initialization of variables
100     OK$="OK"+CHR$(13)
120     NOTOK$="NOTOK"+CHR$(13)
140     ACK$="ACK"+CHR$(13)
160     LOOP=0
180     STARTFLAG=0
200     STARTBATH=0
220     STARTTIME$="00:00:00"
240     JAR.ASIDE.FLAG=0
280     DIM DAYS(12),TESTW#(5),JAR(30),LAST$(5)
300     DAYS(1)=31:DAYS(2)=28:DAYS(3)=31:DAYS(4)=30:DAYS(5)=31:DAYS(6)=30
320     DAYS(7)=31:DAYS(8)=31:DAYS(9)=30:DAYS(10)=31:DAYS(11)=30:DAYS(12)=31

340     ;Initialization of communication ports and data files
360     OPEN "COM1:1200,E,7,1,RS,DSO,LF" AS #1

380     OPEN "STUDIES.DAT" AS #2 LEN=53
400     FIELD #2, 3 AS BATH$, 7 AS IW$, 8 AS LTIME$, 10 AS LDATE$, 7 AS
        INTR$, 8 AS NTIME$, 10 AS NDATE$

        ;Data file STUDIES.DAT, length=53
        ;Purpose: Keeps track of current specimens undergoing immersion
        ;testing

        ;BATH$       Bath location                   3       Integer
        ;IW$         Initial weight                  7       Real
        ;LTIME$      Last weighing time              8       ##:##:##
        ;LDATE$      Last weighing date              10      ####:##:##
        ;INTR$       Current time interval           7       ##:##:##
        ;NTIME$      Next weighing time              8       ##:##:##
        ;NDATE$      Next weighing date              10      ####:##:##

420     OPEN "JARSTATS.DAT" AS #3 LEN=13
440     FIELD #3, 2 AS RATING$, 2 AS FU$, 7 AS CI$, 2 AS WWO$

        ;Data file JARSTATS.DAT, length=13
        ;Purpose: Status rating for all jars in water bath

        ;RATING$     1= 0 to 24 hr weighing intervals 2      Integer
        ;            2= 25 + hr weighing intervals
        ;FU$         jar in use/not in use           2       1/0
        ;CI$         current time interval           7       ####:##
        ;WWO$        jar with/with out water         2       1/0

460     OPEN "STATUS.REP" AS #5 LEN=128
480     FIELD #5, 2 AS TYPE$, 3 AS B$, 7 AS T$, 116 AS MESS$

        ;Data file STATUS.REP, length=128
        ;Purpose: Error diagnostics data file

        ;TYPE$       Type of error, fatal/non-fatal  2       0/1
```

```
        ;B$           Bath location #                         3       Integer
        ;T$           Time of error                           7       ##:##:##
        ;MESS$        Error Message                           116     String

500     LAST(2)=53
520     LAST(3)=13
540     LAST(5)=128
560     PRINT #1, "ENQ"+CHR$(13)
580     GOSUB 5680 ;Wait for controller to acknowledge

600     ;Initialization of Function keys
620     ON KEY(10) GOSUB 6480 ;Stop program and release robot control
640     ON KEY(9) GOSUB 6540 ;Stop program, do not release robot control
660     ON KEY(8) GOSUB 15380 ;Plot data
680     ON KEY(7) GOSUB 15760 ;Start new specimen
700     ON KEY(6) GOSUB 21640 ;Initialize data file JARSTATS.DAT
720     ON KEY(5) GOSUB 14520 ;Initialize/reinitialize Jar array
740     ON KEY(4) GOSUB 22940 ;Print status report
760     KEY(10) ON
780     KEY(9) ON
800     KEY(8) ON
820     KEY(7) ON
840     KEY(6) ON
860     KEY(5) ON
880     KEY(4) ON
900     KEY(10) OFF
920     KEY(9) OFF
940     KEY(8) OFF
960     KEY(7) OFF
980     KEY(6) OFF
1000    KEY(5) OFF
1020    KEY(4) OFF

1021    ;Function Key routines
1022        ;Key (10)—>Terminate program and release controller
1023    ;Key (9) —>Terminate program
1024    ;Key (8) —>Plotting/report generation
1025    ;Key (7) —>Start new specimen
1026    ;Key (6) —>Initialize JARSTATS.DAT
1027    ;Key (5) —>Initialize/Reinitialize Jar(?) array
1028    ;Key (4) —>Print Error Messages
```

;MAIN MODULE

```
1080    ;Determine if specimen(s) need weight measurements taken
1140    LOOP=LOOP+1
1160    ;Open STUDIES.DAT file and test next weight date and time of each
        specimen
1220    IF LOF(2)=0 GOTO 1460 ;Test function keys
1240    FOR I=1 TO (LOF(2)/LAST(2))
1260       GET #2,I
1280       CDT$=DATE$ ;Current date
1300       CDATE$=(MID$(CDT$,7,4))+":"+(MID$(CDT$,1,3))+":"(MID$(CDT$,4,2))
           ;Format Current date: Year:Month:Day
1320       CTM$=TIME$ ;Current time
1340       IF CDATE$>NDATE$ THEN GOSUB 1920 ;Take weight measurement
1360       IF (CDATE$=NDATE$) AND (CTM$>=NTIME$) THEN GOSUB 1920
1380    NEXT I

1420    ;Test function keys
1460    KEY(10) ON
1480    KEY(9) ON
1500    KEY(8) ON
1520    KEY(7) ON
1540    KEY(6) ON
1560    KEY(5) ON
1580    KEY(4) ON
1600    KEY(10) STOP
1620    KEY(9) STOP
1640    KEY(8) STOP
1660    KEY(7) STOP
1680    KEY(6) STOP
1700    KEY(5) STOP
1720    KEY(4) STOP

1740    ;Test for a new specimen waiting to be initialized
1760    ;Proceed with initialization if appropriate
1800    IF (STARTFLAG=1) AND (TIME$>STARTTIME$) THEN GOSUB 19640:LOOP=0:GOTO
        1080
1820    IF LOOP=2 THEN GOSUB 14840:LOOP=0 ;If nothing to do, fill empty jars
1840    GOTO 1080 ;Restart Main Module
1860    CLOSE #1,#2,#3,#4,#5: END
```

;IMMERSION TEST SUBROUTINE

```
1920    ;Immersion Testing procedure
2040    ;Take weight measurement of specimen
2080    VARNAME$="BATH.LOCATION"
2100    BATH=VAL(BATH$)
2120    VARVALUE=BATH
2140    GOSUB 6000 ;Set BATH.LOCATION to BATH in controller
2180    VARNAME$="WEIGHT.TYPE"
2200    VARVALUE=1
2220    GOSUB 6000 ;Set WEIGHT.TYPE to 1 in controller
2260    IF JAR(BATH)=1 GOTO 1380 ;If error flag set, then move on to next jar
2280    GOSUB 6720 ;Take weight measurement
2300    VARNAME$="MW.SAMPLE"
2320    GOSUB 6360 ;Get value of MW.SAMPLE (weight of specimen)
2360    WEIGHT#=VARVALUE

2400    ;Locate specimen data file and update information
2440    OPEN "DATAFILE.DAT" AS #4 LEN=12
2460    FIELD #4, 12 AS FILES$

        ;Data file DATAFILE.DAT, length=12
        ;Purpose: Lists file names associated with each Jar in bath

        ;FILES$      File name and extension          12     String

2480    GET #4,BATH
2500    CLOSE #4

2540    ;Open data file
2580    OPEN FILES$ AS #4 LEN=48
2600    FIELD #4, 10 AS DDATE$, 8 AS DTIME$, 10 AS DHRS$, 10 AS DWEIGHT$, 10
        AS DCHW$

        ;Data file *file*.DAT, length=48
        ;Purpose: Unique data file for each specimen

        ;DDATE$      Date of weighing                 10     ##:##:####
        ;DTIME$      Time of weighing                 8      ##:##:##
        ;DHRS$       Total hours in study             10     real
        ;DWEIGHT$    Weight of specimen               10     real
        ;DCHW$       Change in weight                 10     real

2620    GET #4,(LOF(4)/48)
2640    CHRS=VAL(DHRS$) ;Hours in study
2660    CDT$=DATE$ ;Current date
2680    CDATE$=(MID$(CDT$,7,4))+"-"+(MID$(CDT$,1,3))+"-"+(MID$(CDT$,4,2))
        ;Reformat date
2700    CHW=WEIGHT#-VAL(IW$)

        ;Set date, time, weight and change in weight of specimen
2740    LSET DDATE$=CDT$
2760    LSET DTIME$=CTM$
2780    RSET DWEIGHT$=STR$(WEIGHT#)
```

```
2800    RSET DCHW$=STR$(CHW)

2820    ;Get last weighing time from file
2840    LH=VAL(MID$(LTIME$,1,2)) ;Hours
2860    LM=VAL(MID$(LTIME$,4,2)) ;Minutes
2880    LS=VAL(MID$(LTIME$,7,2)) ;Seconds

        ;Get current time
2900    CH=VAL(MID$(CTM$,1,2))
2920    CM=VAL(MID$(CTM$,4,2))
2940    CS=VAL(MID$(CTM$,7,2))

2980    ;Calculate number of hours in current study
3020    IF CDATE$=LDATE$ GOTO 3800 ;If current day and the last day a
        weighing was made are equal

3040    ;Calculate number of hours in days past
3060    LD=VAL(MID$(LDATE$,9,2)) ;Last day
3080    CD=VAL(MID$(CDATE$,9,2) ;Current day
3100    LMO=VAL(MID$(LDATE$,6,2) ;Last month
3120    CMO=VAL(MID$(CDATE$,6,2) ;Current month

3160    TH=23: TM=59: TS=59
3220    THRS=(TH-LH)+(TM-LM)/60+(TS-LS)/3600 ;Number of hours left in last
        day a weighing was made
3240    IF (MID$(CDATE$,1,4))=(MID$(LDATE$,1,4)) GOTO 3560 ;If the year has
        not changed
3280    THRS=THRS+24*(DAYS(LMO)-LD) ;Number of hours left in the last month a
        weighing was made
3300    TMO=LMO+1 ;next month

        ;Calculate number of hours left in the remainder of the year of the
        last weighing made
3320    WHILE TMO<=12
3340       THRS=THRS+24*(DAYS(TMO))
3360       TMO=TMO+1
3380    WEND

3400    ;Calculate number of hours in current year past
3420    TMO=1
3440    WHILE TMO<CMO ;While calculated month is less than current month
3460       THRS=THRS+24*(DAYS(TMO))
3480       TMO=TMO+1
3500    WEND

        ;Calculate number of hours past in current month
3520    THRS=THRS+24*(CD-1)
3540    GOTO 3940 ;Calculation finished

        ;Year has not changed
3560    IF CMO=LMO GOTO 3740
3580    THRS=THRS+24*(days(LMO)-LD) ;Hours left in month of last weighing
3600    TMO=LMO+1
3620    WHILE TMO<CMO
```

```
3640       THRS=THRS+24*(DAYS(TMO))
3660       TMO=TMO+1
3680    WEND
3700    THRS=THRS+24*(CD-1)
3720    GOTO 3940 ;Calculation finished

        ;Month has not changed
3740    THRS=THRS+24*(CD-LD-1) ;Days past in current month
3760    GOTO 3940 ;Calculation finished

3780    ;Day has not changed
3800    IF CS<LS THE CS=CS+60: CM=CM-1
3820    IF CS>=LS THEN DS=CS-LS
3840    IF CM<LM THEN CM=CM+60: CH=CH-1
3860    IF CM>=LM THEN DM=CM-LM
3880    DH=CH-LH
3900    THRS=DH+DM/60+DS/3600 ;Calculation finished

3920    ;Store new information
3940    RSET DHRS$=STR$(THRS+CHRS)
3960    PUT #4, (LOF(4)/48+1)

3980    ;Calculate next weighing time and date
4000    ;Increment time interval between weighings if appropriate
4060    IH=VAL(MID$(INSTR$,1,4))
4080    IM=VAL(MID$(INSTR$,6,2))
4100    NH=VAL(MID$(CTM$,1,2))
4120    NM=VAL(MID$(CTM$,4,2))
4140    NS=VAL(MID$(CTM$,7,2))
4160    NY=VAL(MID$(CDATE$,1,4))
4180    NMO=VAL(MID$(CDATE$,6,2))
4200    ND=VAL(MID$(CDATE$,9,2))
4220    CT=(LOF(4)/48-4) ;Looking at last five weight measurements
4240    IF CT<1 GOTO 4640 ;First weighing
4260    C=1
4280    TESTAV#=0
4300    FOR CC=CT TO (LOF(4)/48)
4320       GET #4, CC
4340       TESTW#(C)=VAL(DWEIGHT$)
4360       TESTAV#=TESTAV#+TESTW#(C)
4380       C=C+1
4400    NEXT CC
4420    AW#=TESTAV#/5
4440    FLAG=0
4460    FOR C=1 TO 5
4480       IF ABS(TESTAV#-TESTW#(C))>0.0002 THEN FLAG=FLAG+1
4500    NEXT C
4520    IF FLAG>1 GOTO 4640 ;Do not increase weight time increment
4540    TM=IM+IM:TH=0 ;Double time increment
4560    IF TM>=60 THEN TH=TH+1:TM=TM-60
4580    TH=TH+IH+IH
4600    IH=TH
4620    IM=TM
4640    GET #3,BATH
```

```
4660    TT$=STR$(IH):TT=LEN(TT$)-1:IH$=MID$(TT$,2,TT)
4680    ON TT GOSUB 4720,4740,4760,4770
4700    GOTO 4780
4720    IH$="000"+IH$: RETURN
4740    IH$="00"+IH$: RETURN
4760    IH$="0"+IH$: RETURN
4770    RETURN
4780    TT$=STR$(IM):TT=LEN(TT$)-1:IM$=MID$(TT$,2,TT)
4800    IF TT=1 THEN IM$="0"+IM$
4820    RSET CI$=IH$+":"+IM$
4840    PUT #3,BATH

4860    LSET LTIME$=CTM$
4880    LSET LDATE$=CDATE$
4900    RSET INTR$=IH$+":"+IM$
4920    TM=NM+IM: TH=0
4940    IF TM>=60 THEN TH=TH+1: TM=TM-60
4960    TH=TH+NH+IH
4980    IF TH<=23 THEN NH=TH: NM=TM: GOTO 5180

5000    WHILE (TH>23)AND(ND<DAYS(NMO))
5020       ND=ND+1
5040       TH=TH-24
5060    WEND
5080    IF TH<=23 GOTO 5180
5100    NMO=NMO+1
5120    ND=0
5140    IF NMO>12 THEN NY=NY+1:NMO=1
5160    GOTO 5000

5180    TT$=STR$(NY): TT=LEN(TT$)-1: NY$=MID$(TT$,2,TT)
5200    TT$=STR$(NMO): TT=LEN(TT$)-1: NMO$=MID$(TT$,2,TT)
5220    IF TT=1 THEN NMO$="0"+NMO$
5240    TT$=STR$(ND): TT=LEN(TT$)-1: ND$=MID$(TT$,2,TT)
5260    IF TT=1 THEN ND$="0"+ND$
5280    TT$=STR$(NH): TT=LEN(TT$)-1: NH$=MID$(TT$,2,TT)
5300    IF TT=1 THEN NH$="0"+NH$
5320    TT$=STR$(NM): TT=LEN(TT$)-1: NM$=MID$(TT$,2,TT)
5340    IF TT=1 THEN NM$="0"+NM$
5360    TT$=STR$(NS): TT=LEN(TT$)-1: NS$=MID$(TT$,2,TT)
5380    IF TT=1 THEN NS$="0"+NS$
5400    LSET NDATE$+"-"+NMO$+"-"+ND$
5420    LSET NTIME$=NH$+":"+NM$+":"+NA$
5440    PUT #2,I
5460    CLOSE #4
5480    RETURN
```

```
                ;CONTROLLER SUBROUTINES

5640    ;Controller Ready?
5680    MESSAGE$=""
5700    WHILE (MESSAGE$<>ACK$)
5720       GOSUB 5840
5740    WEND
5760    RETURN

5800    ;Aquire messages from controller
5840    MESSAGE$=""
5860    CHARIN$=INPUT$(1,#1)
5880    IF CHARIN$=CHR$(10) THEN RETURN
5900    MESSAGE$=MESSAGE$+CHARIN$
5920    GOTO 5860

5960    ;Set the variable VARNAME to the value of VARVALUE in the controller
6000    PRINT #1, VARNAME$+"="+STR$(VARVALUE)+CHR$*(13)
6020    GOSUB 6120
6040    RETURN

6080    ;Wait for response from controller
6120    MESSAGE$=""
6140    WHILE (MESSAGE$<>OK$)AND(MESSAGE$<>NOTOK$)
6160       GOSUB 5840
6180    WEND

6220    ;Test if communications problem
6260    IF MESSAGE$=CHR$(10)+"NOTOK" THEN RETURN 1860
6280    RETURN

6320    ;Retrieve value of VARNAME from controller
6360    PRINT #1, "? "+VARNAME$+CHR$(13)
6380    GOSUB 5840
6400    VARVALUE=VAL(MID$(MESSAGE$,3,LEN(MESSAGE$)-1))
6420    GOSUB 6120
6440    RETURN

6480    ;Stop program and disconnect communications
6500    PRINT #1, "BYE"+CHR$(13): CLOSE #1,#2,#3,#4,#5: RETURN 1860

6540    ;Stop program, do not disconnect communications
6560    CLOSE #1,#2,#3: RETURN 1860
```

;CONTROL SUBROUTINES FOR RUNNING IMMERSION TEST

```
6720    ;Take weight measurement
6760    GOSUB 7160 ;Get appropriate specimen jar, and uncap jar
6780    GOSUB 7480 ;Get specimen from jar
6800    GOSUB 7960 ;Blot specimen
6820    GOSUB 8260 ;Make weight measurement, return specimen to jar
6840    PROGNAME$="DRYER.ON" ;dry blotter
6860    GOSUB 9060 ;Exicute program
6880    GOSUB 8640 ;Cap jar and return to water bath
6900    RETURN

6940    ;Initialize a new specimen
6980    GOSUB 7140 ;Get appropriate specimen jar, and uncap jar
7000    GOSUB 7480 ;Get new specimen from specimen holder
7020    GOSUB 8260 ;Take initial weight measurement, return specimen to jar
7040    CTM$=TIME$ ;Record initial time
7060    GOSUB 8640 ;Cap jar and return to water bath
7080    RETURN

7120    ;Get appropriate specimen jar, and uncap jar
7160    PROGNAME$="REMOVE.JAR"
7180    GOSUB 9060 ;Exicute program in controller
7200    VARNAME$="JAR.FLAG" ;Error flag for gripping jar properly
7220    GOSUB 6360 ;Get value of JAR.FLAG
7240    IF VARVALUE>0 THEN GOSUB 9260 ;Regrip jar properly
7260    PROGNAME$="UNCAP.JAR"
7280    GOSUB 9060
7300    VARNAME$="PHOTO.SENSOR" ;Error flag for detecting jar
7320    GOSUB 6360
7340    IF VARVALUE=1 THEN GOSUB 10640 ;Cap not found, try uncapping again
7360    PROGNAME$="CAP.ASIDE"
7380    GOSUB 9060
7400    RETURN

7440    ;Get specimen from jar
7480    PROGNAME$="GET.SAMPLE.FROM.JAR"
7500    GOSUB 9060
7520    VARNAME$="PHOTO.SENSOR.2"
7540    GOSUB 6360
7560    IF VARVALUE=0 THEN GOSUB 11160 ;Jar lifted with specimen
7580    PROGNAME$="CHECK.FOR.SAMPLE.FROM.JAR"
7600    GOSUB 9060
7620    VARNAME$="SAMPLE.FLAG"
7640    GOSUB 6360
7660    IF VARVALUE=1 THEN GOSUB 11380 ;sample not retrieved from jar
7680    CTM$=TIME$ ;Record initial time of study
7700    RETURN

7740    ;Get new specimen from holder
7780    PROGNAME$="GET.SPECIMEN"
7800    GOSUB 9060
7820    VARNAME$="SAMPLE.FLAG"
7840    GOSUB 6360
```

```
7860    IF VARVALUE=1 THEN GOSUB 14220 ;No specimen in holder
7880    RETURN

7920    ;Blot specimen
7960    CTM$=TIME$ ;Record time of weight measurement
7980    PROGNAME$="BLOT.SAMPLE"
8000    GOSUB 9060
8020    VARNAME$="ERROR.BLOT"
8040    GOSUB 6360
8060    IF VARVALUE>0 THEN GOSUB 11840 ;Blotter malfunction
8080    PROGNAME$="CHECK.FOR.SAMPLE.FROM.BLOTTER"
8100    GOSUB 9060
8120    VARNAME$="SAMPLE.FLAG"
8140    GOSUB 6360
8160    IF VARVALUE=1 THEN GOSUB 12640 ;Sample dropped during blotting
8180    RETURN

8220    ;Take weight measurement, return specimen to jar
8260    PROGNAME$="TAKE.WEIGHT"
8280    GOSUB 9060
8300    VARNAME$="ERROR.BALANCE"
8320    GOSUB 6360
8340    IF VARVALUE>0 THEN GOSUB 12780 ;balance malfunction
8360    PROGNAME$="CHECK.FOR.SAMPLE.FROM.BALANCE"
8380    GOSUB 9060
8400    VARNAME$="SAMPLE.FLAG"
8420    GOSUB 6360
8440    IF VARVALUE=1 THEN GOSUB 13360 ;Sample dropped in balance
8460    PROGNAME$="PUT.SAMPLE.IN.JAR"
8480    GOSUB 9060
8500    VARNAME$="PHOTO.SENSOR.2"
8520    GOSUB 6360
8540    IF VARVALUE=0 THEN GOSUB 13520 ;jar lifted by forceps
8560    RETURN

8600    ;Cap jar and return to water bath
8640    PROGNAME$="GET.CAP"
8660    GOSUB 9060
8680    VARNAME$="PHOTO.SENSOR"
8700    GOSUB 6360
8720    IF VARVALUE=1 THEN GOSUB 13660 ;cap not retrieved from holder
8740    PROGNAME$="CAP.JAR"
8760    GOSUB 9060
8780    VARNAME$="PHOTO.SENSOR.2"
8800    GOSUB 6360
8820    IF VARVALUE=0 THEN GOSUB 13900 ;jar not capped properly
8840    PROGNAME$="GET.JAR"
8860    GOSUB 9060
8880    VARNAME$="JAR.FALG"
8900    GOSUB 6360
8920    IF VARVALUE>0 THEN GOSUB 9280 ;Regrip jar
8940    PROGNAME$="REPLACE.JAR"
8960    GOSUB 9060
8980    RETURN
```

```
9020    ;Run program PROGNAME$ in controller
9060    PRINT #1, PROGNAME$+CHR$(13)
9080    GOSUB 6100
9100    RETURN
```

```
                    ;ERROR HANDLING ROUTINES

9280    ;Jar not detected comming from water bath
9329    COUNT=0
9360    ;Try regripping jar no more than three times
9400    WHILE (VARVALUE>0) AND (COUNT<3)
9420       COUNT=COUNT+1
9440       MESSAGE$=STR$(VARVALUE)+" SIDE(S) OF SAMPLE JAR UNDETECTED."
9460       IF VARVALUE=4 GOTO 9920
9480       MESSAGE$=MESSAGE$+" REGRIPPING PROCEDURE EXICUTED.": GOSUB 10040
9520       RSET TYPE$="0": GOSUB 14360 ;Print message in STATUS.REP
9540    WEND
9580    IF VARVALUE=0 THEN RETURN ;Resume procedure if error corrected
9600    ON VARVALUE GOTO 9620,9800,9800,9920
9620    RSET TYPE$="1"
9640    MESSAGE$="REGRIPPING PROCEDURE UNSUCCESSFUL."

        ;reset error flag JAR.ASIDE.FLAG, flag specimen jar, print error
        message, and move on to next jar
9680    IF JAR.ASIDE.FLAG=0 THEN MESSAGE$=MESSAGE$+" JAR SET ASIDE.":
        JAR.ASIDE.FLAG=1: JAR(BATH)=1: GOSUB 14360: RETURN 1400

9720    MESSEAGE$=MESSAGE$+" NO ROOM TO PUT JAR ASIDE, SYSTEM SHUT DOWN
        PROCEDURE EXICUTED."
9740    GOSUB 14360 ;Print message
9760    GOSUB 10200 ;System shut down procedure
9780    RETURN 1860 ;Terminate program

9800    RSET TYPE$=1
9820    MESSAGE$="JAR MAY NEED REPAINTING. REPLACING JAR IN BATH ATTEMPTED."
9840    GOSUB 10360 ;return jar to water bath
9860    JAR(BATH)=1 ;flag jar (i.e. discontinue study)
9880    GOSUB 14360 ;print message
9900    RETURN 1400 ;Next jar

9920    IF COUNT=1 THEN MESSAGE$=MESSAGE$+" JAR NOT PRESENT OR
        UNRETRIEVABLE": RSET TYPE$="1": GOSUB 14360: JAR(BATH)=1: RETURN 1400

9960    MESSAGE$="JAR DROPPED DURING REGRIPPING PROCEDURE, SYSTEM SHUT DOWN
        PROCEDURE EXICUTED.": GOSUB 14360: JAR(BATH)=1: GOSUB 10200

10000   RETURN 1860 ;Terminate program


10040   ;Regripping procedure
10080   PROGNAME$="REGRIPPING"
10100   GOSUB 9060
10120   VARNAME$="JAR.FLAG"
10140   GOSUB 6360
10160   RETURN

10220   ;System shut down
10260   PROGNAME$="DRYER.OFF"
10280   GOSUB 9060
```

```
10300   PRINT#1, "BYE"+CHR$(13)
10320   RETURN 1860

10360   ;Put jar in water bath
10400   PROGNAME$="PUT.JAR.IN.BATH"
10420   GOSUB 9060
10440   RETURN

10480   ;Get cap from holder
10520   PROGNAME$="AVOID.SENSOR": GOSUB  9060
10540   PROGNAME$="GET.CAP.FROM.HOLDING": GOSUB 9060
10560   PROGNAME$="AVOID.SENSOR": GOSUB 9060
10580   PROGNAME$="MOVE.TO.CAPPING.CAP": GOSUB 9060
10600   RETURN

10660   ;Cap not removed from jar properly
10700   COUNT=0
10720   WHILE (VARVALUE=1) AND (COUNT<3)
10740      COUNT=COUNT+1
10760      MESSAGE$="JAR UNCAPPED IMPROPERLY.  UNCAPPING PROCEDURE REPEATED."
10780      RSET TYPE$="0"
10800      GOSUB 14360
10820      PROGNAME$="GP.HAND": GOSUB 9060
10840      GOSUB 11000; Uncap jar
10860   WEND
10880   IF VARVALUE-0 THEN RETURN
10900   MESSAGE$="UNCAPPING UNSUCCESSFUL.  JAR REPLACED IN BATH."
10920   RSET TYPE$="1"
10940   GOSUB 14360; print message
10960   GOSUB 8840; Get jar, put jar in bath
10980   RETURN 1400 ;next jar

11020   ;Uncapping procedure
11060   PROGNAME$="UNCAP.JAR"
11080   GOSUB 9060
11100   VARNAME$="PHOTO.SENSOR"
11120   GOSUB 6360
11140   RETURN

11180   ;Jar lifted with sample
11220   MESSAGE$="JAR LIFTED WHILE TRYING TO RETRIEVE SPECIMEN FROM JAR.  JAR
        REPLACED IN BATH.": GOSUB 14360
11240   RSET TYPE$="1"
11260   PROGNAME$="JAR.HAND1"
11280   GOSUB 9060
11300   GOSUB 8640; Get cap, cap jar, put jar in bath
11320   JAR(VAL(BATH$))=1
11340   RETURN 1400; Next jar

11380   ;sample not detected from jar
11420   MESSAGE$-"SPECIMEN NOT RETECTED DURING REMOVAL ATTEMPT FROM JAR.
        PROCEDURE REPEATED."
11440   RSET TYPE$="0"
11460   GOSUB 14360; Print message
```

```
11480   GOSUB 8460; Put sample in jar
11500   PROGNAME$="GET.SAMPLE.FROM.JAR"
11520   GOSUB 9060
11540   PROGNAME$="CHECK.FOR.SAMPLE.FROM.JAR"
11560   GOSUB 9060
11580   VARNAME$="SAMPLE.FLAG"
11600   GOSUB 6360
11620   IF VARVALUE=0 THEN RETURN
11640   MESSAGE$=MESSAGE$+"SPECIMEN RETRIEVAL UNCSUCCESSFUL. JAR REPLACED IN
        BATH."
11660   RSET TYPE$="1":GOSUB 14360
11680   GOSUB 8460 ;Put sample in jar
11700   GOSUB 8640 ;Get cap, cap jar, put jar in bath
11720   JAR(BATH)=1
11740   RETURN 1400 ;Next jar


11800   ;Blotter malfunction
11840   MESSAGE$="BLOTTER MALFUNCTION: "
11860   ON VARVALUE GOSUB 12020,12060
11880   MESSAGE$=MESSAGE$+"SUTDOWN PROCEDURE EXICUTED. "
11900   RSET TYPE$="1"
11920   GOSUB 14360 ;Print message
11940   ON VARVALUE GOTO 12180,1860
11960   RETURN

12020   MESSAGE$=MESSAGE$+"WILL NOT CLOSE. "
12040   RETURN

12060   MESSAGE$=MESSAGE$+"WILL NOT OPEN. "
12080   RETURN

12140   ;Determining type of malfunction
12180   VARNAME$="HOLD.COUNT"
12200   GOSUB 6360
12220   PROGNAME$="ERROR.BLOT.ROUTINE"
12240   GOSUB 9060
12260   ON VARVALUE GOSUB 1244,12300,12360
12280   RETURN 1860

12300   PROGNAME$="PUT.SAMPLE.IN.HOLDER.2"
12320   GOSUB 9060
12340   GOTO 12400

12360   PROGNAME$="PUT.SAMPLE.IN.HOLDER.3"
12380   GOSUB 9060
12400   PROGNAME$="GET.SAMPLE.FROM.HOLDER.1"
12420   GOSUB 9060
12440   GOSUB 8460 ;Put sample in jar
12460   GOSUB 8640; Get cap, cap jar, put jar in bath
12480   GOSUB 10200; System Shutdown
12500   RETURN

12560   ;Blotter will not close
```

```
12600   RETURN 1860; Terminate program

12660   ;Sample dropped during blotting
12700   MESSAGE$=MESSAGE$+"SAMPLE NOT DETECTED AFTER BLOTTING. JAR PUT BACK
        IN BATH. "
12720   RSET TYPE$="1": GOSUB 14360
12740   GOSUB 8640
12760   RETURN 1400

12800   ;Balance door malfunction
12840   MESSAGE$="BALANCE DOOR MALFUNCTION: "
12860   ON VARVALUE GOSUB 12960,13040,13140,13280
12880   RSET TYPE$="0"
12900   GOSUB 8640
12920   GOSUB 10220
12940   RETURN 1860

12980   MESSAGE$=MESSAGE$+"BALANCE DOOR WILL NOT OPEN; ERROR 1. SAMPLE AND
        JAR PUT BACK IN BATH"
13000   GOSUB 8460
13020   RETURN

13060   MESSAGE$=MESSAGE$+"BALANCE DOOR WILL NOT OPEN: ERROR 2. JAR PUT BACK
        IN BATH"
13080   PROGNAME$="MOVE TO CAPPING.FORCEPS"
13100   GOSUB 9060
13120   RETURN

13160   MESSAGE$=MESSAGE$+"BALANCE DOOR WILL NOT CLOSE: ERROR 3. WEIGHT NOT
        RECORDED, SAMPLE AND JAR BUT BACK IN BATH"
13180   PROGNAME$="SAMPLE.IN.BALANCE.OUT":GOSUB 9060
13200   PROGNAME$="GRIP.SAMPLE":GOSUB 9060
13220   PROGNAME$="MOVE.TO.BALANCE.OUT":GOSUB 9060
13240   GOSUB 8460
13260   RETURN

13300   MESSAGE$=MESSAGE$+"BALANCE DOOR WILL NOT CLOSE: ERROR 4. WEIGHT
        RECORDED, SAMPLE AND JAR RETURNED TO BATH"
13320   GOSUB 8460
13340   RETURN

13380   ;Sample dropped in balance
13420   MESSAGE$="SAMPLE DROPPED IN BALANCE."
13440   RSET TYPE$="1":GOSUB 14360
13460   PROGNAME$="MOVE.TOWARDS.BLOTTER": GOSUB 20850
13480   GOSUB 8640
13500   RETURN 1400

13540   ;Jar lifted with forceps
13580   MESSAGE$="JAR LIFTED WITH FORCEPS AS SAMPLE IS PLACED IN JAR."
13600   RSET TYPE$="0":GOSUB 14360
13620   PROGNAME$="JAR.HANG2":GOSUB 9060
13640   RETURN
```

```
13680   ;Cap missing from holder
13720   MESSAGE$="CAP MISSING FROM HOLDER. WILL GET EXTRA CAP IF AVAILABLE. "
13740   PROGNAME$="NEXT.CAP":GOSUB 9060
13760   VARNAME$="PHOTO.SENSOR":GOSUB 6360
13780   IF VARVALUE=0 THEN RSET TYPE$="0": GOSUB 14360:RETURN
13800   IF JAR.ASIDE.FLAG=1 THEN MESSAGE$=MESSAGE$+"NO EXTRA CAP FOUND. ASIDE
        POSITION FULL. SYSTEM SHUT DOWN.":RSET TYPE$="1":GOSUB 14360:GOSUB
        10260:RETURN 1860
13820   MESSAGE$=MESSAGE$+"NO EXTRA CAP FOUND, UNCAPPED JAR PUT ASIDE."
13840   RSET TYPE$="1":GOSUB 14360
13860   PROGNAME$="UNCAPPED.JAR.ASIDE":JAR.ASIDE.FLAG=1:GOSUB 9060
13880   RETURN 1400

13920   ;Jar not capped properly
13960   MESSAGE$="JAR NOT THREADED PROPERLY DURING CAPPING PROCEDURE.
        UNCAPING/RECAPPING ATTEMPTED."
13980   RSET TYPE$="0": GOSUB 14360
14000   GOSUB 11020
14020   IF VARVALUE=1 THEN GOSUB 10700; Jar not uncapped
14040   PROGNAME$="CAP.ASIDE":GOSUB 9060
14060   PROGNAME$="GET.CAP":GOSUB 9060
14080   VARNAME$="PHOTO.SENSOR":GOSUB 6360
14100   IF VARVALUE=0 THEN RETURN
14120   MESSAGE$="RECAPPING PROCEDURE UNCSUCCESSFUL. "
14140   RSET TYPE$="1"
14160   IF JAR.ASIDE.FLAG=1 THEN MESSAGE$=MESSAGE$+"NO AVAILABLE POSITIONS TO
        PLACE JAR ASIDE. SYSTEM SHUT DOWN.": GOSUB 14360:GOSUB 10220:RETURN
        1860
14180   RETURN

14380   ;Print Message to STATUS.REP
14420   RSET B$=STR$(BATH)
14440   LSET T$=TIME$
14460   LSET MESS$+MESSAGE$
14480   PUT #5, (LOF(5)/128)+1
14500   RETURN
```

;FUNCTION KEY ROUTINES

```
14520    ;Initialize jar(?) Array (Function key 5)
14580    LOOP=0
14600    PRINT:PRINT "INITIALIZING STATUS OF SAMPLE JARS IN WATER BATH":PRINT
14620    INPUT "INITIALIE ALL JARS? (Y/N): ",ANS$
14640    IF ANS$="N" THEN GOTO 14740
14660    FOR C=1 TO 30 ;Number of jar locations in bath
14680       JAR(C)=0
14700    NEXT C
14720    RETURN

14740    INPUT "ENTER JAR #,VALUE # (0/1) $ENTER 0,0 TO END†: ",K,KV
14760    IF K=0 THEN RETURN
14780    JAR(K)=KV
14800    GOTO 14740

14840    ;Fill Empty jars
14880    FOR FILL=1 TO (LOF(3)/13)
14900       GET #3,FILL
14920       JR=VAL(RATING$)
14940       IF JR=0 GOTO 15020
14960       BATH=FILL
14980       JWWO=VAL(WWO$)
15000       IF JWWO=0 THEN GOSUB 15060:RSET WWO$="1":PUT #3,BATH: GOTO 15040
15020    NEXT FILL
15040    RETURN

15060    ;Filling Jars
15120    VARNAME$="BATH.LOCATION"
15140    VARVALUE=BATH: GOSUB 6000
15160    GOSUB 7120; Get jar, uncap jar, put cap aside
15180    PROGNAME$="GET.JAR.FROM.CAPPING.NO.CAP":GOSUB 9060
15200    PROGNAME$="FILL.JAR":GOSUB 9060
15220    PROGNAME$="PUT.JAR.IN.CAPPING.NO.CAP":GOSUB 9060
15240    GOSUB 10460:GOSUB 8680; Cap jar, put jar in bath
15260    RETURN

15320    ;Plot data (Function key 8)
15380    INPUT "ENTER FILE TO BE PLOTTED ",FILE$
15400    CLOSE:OPEN FILE$ FOR INPUT AS #1
15420    OPEN "RS1.TMP" FOR OUTPUT AS #2
15440    FOR I=1 TO 6:INPUT #1, DUMMY%: NEXT I
15460    INPUT #1, IWEIGHT
15480    FOR I=1 TO 6: INPUT #1,DUMMY$:NEXT I
15500    INPUT #1, DUM: INPUT #1,DUM
15520    WHILE NOT EOF(1)
15540       INPUT #1,ITIME,WEIGHT
15560       STIME=SQR(ITIME)
15580       SWEIGHT=100*((WEIGHT-IWEIGHT)/IWEIGHT)
15600       PRINT #2,USING "###.####"; SWEIGHT;
15620       PRINT #2,",";
15640       PRINT #2,STIME
15660    WEND
```

```
15680    LOOP=0
15700    CLOSE:RETURN 1860
```

;STARTING A NEW STUDY (OR SPECIMEN) (FUNCTION KEY 7)

```
15760     LOOP=0
15840     ONE=0:TWO=0
15880     ;Search for available jar with rating=1 (jar filled with water)
15900     FOR C=1 TO (LOF(3)/LAST(3))
15920        GET #3,C
15940        JR=VAL(RATING$)
15960        IF JR<>1 GOTO 16120 ;Don't consider jars with a rating=2
15980        JFU=VAL(FU$)
16000        JH=VAL(MID$(CI$,1,4))
16020        JM=VAL(MID$(CI$,6,2))
16040        JWWO=VAL(WWO$)
16060        IF (JFU=0) AND (JWWO=1) THEN BATH=C:GOTO 16260; Jar ready for use
16080        IF (JFU=0) AND (JWWO=0) TTHEN ONE=C:GOTO 16120; Possible jar,
             continue searching
16100        IF (JH>=24) AND (JAR(C)=0) THEN TWO=C; Possible jar for switching
16120     NEXT C
16140     IF ONE>0 THEN GOSUB 19080:GOTO 18940 ;Fill jar 'ONE' with water
16160     IF TWO>0 THEN GOSUB 19900:GOTO 18940 ;Switch jars
16180     PRINT:PRINT "NO JARS AVAILABLE AT THIS TIME...":RETURN

16220     ;User Input for specimen specifications
16260     GET #3,BATH
16280     INPUT "ENTER SAMPLE DESCRIPTION: ",DISC$
16300     INPUT "ENTER RESIN TYPE: ",RESIN$
16320     INPUT "ENTER FIBER TYPE AND NUMBER OF PLY: ",FIBER$
16340     INPUT "ENTER FIBER CONFIGURATION: ",CONF$
16360     INPUT "ENTER TEMPERATUR OF ENVIRONMENT: ",TEMP$
16380     INPUT "ENTER CONDITIONING PARAMETERS (25 CHAR): ",COND$
16400     INPUT "ENTER INITIAL TIME INTERVAL (MIN): ";INTERVAL
16420     PRINT:PRINT
16440     INPUT "ENTER FILE NAME FOR NEW SAMPLE: ",FFILE$
16460     PRINT:INPUT "ENTER SAMPLE THICKNESS: ",THICK$
16480     PRINT:INPUT "ENTER SAMPLE HEIGHT DIMENSION: ",A
16500     PRINT:INPUT "ENTER SAMPLE WIDTH DIMENSION: ",B
16520     PRINT:INPUT "ENTER SAMPLE DIAGONAL (RIGHT TOP TO LEFT BOTTOM): ",C
16540     PRINT:INPUT "ENTER SAMPLE DIAGONAL (LEFT TOP TO RIGHT BOTTOM): ",D
16560     JH=0:JM=INTERVAL
16580     IF INTERVAL >=60 THEN INTR%=INTERVAL/60:JH=INTR%:JM=INTERVAL-(JH*60)
16600     TT$=STR(JH):TT=LEN(TT$)-1:JH$=MID$(TT$,2,TT)
16620     ON TT GOSUB 16660,16680,16700,16640
16640     GOTO 16720

16660     JH$="000"+JH$:RETURN
16680     JH$="00"+JH$:RETURN
16700     JH$="0"+JH$:RETURN

16720     TT$=STR$(JM):TT=LEN(TT$)-1:JM$=MID$(TT$,2,TT)
16740     IF TT=1 THEN JM$="0"+JM$
16760     RSET CI$=JH$+:"+JM$
16780     RSET FU$=STR$(1)
16800     RSET WWO$=STR$(1)
16820     PUT #3,BATH
```

```
16860   ;Calculate sample surface area
16920   AREA=A*B
16940   HYP=SQR(A^2+B^2)
16960   DIAG1=(HYP-C)/2
16980   DIAG2=(HYP-D) 2
17000   AREA1=4*DIAG1
17020   AREA2=4*DIAG2
17040   AREA=AREA-AREA1-AREA2

17120   ;Store inital data in files
17160   OPEN "DATAFILE.DAT" AS #4 LEN=12
17180   FIELD #4, 12 AS FILE$
17200   LSET FILE$=FFILE$
17220   PUT #4, BATH
17240   CLOSE #4

17280   RSET BATH$=STR$(BATH)
17300   RSET INTR$=JH$+":"+JM$
17320   PRINT:PRINT "PLACE SPECIMEN IN SAMPLE HOLDER"

17360   ;Take initial weighing
17400   PRINT:PRINT:PRINT "HIT ANY KEY TO CONTINUE"
17420   A$=INKEY$:IF A$="" THEN 17420
17440   VARNAME$="BATH.LOCATION"
17460   VARVALUE=BATH
17480   GOSUB 6000
17500   VARNAME$="WEIGHT.TYPE"
17520   VARVALUE=0
17540   GOSUB 6000
17560   GOSUB 6940;Take initial weight measurement

17580   ;Calculating and storing time for next interval
17620   CDT$=DATE$
17640   CDATE$=MID$(CDT$,7,4)+"-"+MID$(CDT$,1,3)+"-"+MID$(CDT$,4,2)
17660   LSET LTIME$=CTM$
17680   LSET LDATE$=CDATE$
17700   VARNAME$="IW.SAMPLE(BATH.LOCATION)"
17720   GOSUB 6360
17740   IWEIGHT#=VARVALUE
17760   RSET IW#=STR$(IWEIGHT#)
17780   CH=VAL(MID$(CTM$,1,2))
17800   CM=VAL(MID$(CTM$,4,2))
17820   NH=0
17840   NM=CM+JM
17860   IF NM>=60 THEN NM=NM-60:NH=NH+1
17880   NH=NH+CH+JH
17900   TT$=STR$(NH):TT=LEN(TT$)-1:NH$=MID$(TT$,2,TT)
17920   IF TT=1 THEN NH$="0"+NH$
17940   TT$=STR$(NM):TT=LEN(TT$)-1:NM$=MID$(TT$,2,TT)
17960   IF TTR=1 THEN NM$+"0"+NM$
17980   IF NH<=23 THEN LSET NTIME$=NH$+":"+NM$+":"+MID$(CTM$,7,2):LSET
        NDATE$=CDATE$:GOTO 18400
18000   NMO=VAL(MID$(CDT$,1,2))
18020   ND=VAL(MID$(CDT$,4,2))
```

```
18040     NY=VAL(MID$(CDT$,7,4))
18060     WHILE NH>23
18080        ND=ND+1
18100        IF ND>DAYS(NMO) THEN NMO=NMO+1:ND=1
18120        IF NMO>12 THEN NY=NY+1:NMO=1:ND=1
18140        NH=NH-24
18160     WEND
18180     TT$=STR$(NH):TT-LEN(TT$)-1:NH$=MID$(TT$,2,TT)
18200     IF TT=1 THEN NH$="0"+NH$
18220     TT$=STR$(NM):TT-LEN(TT$)-1:NM$=MID$(TT$,2,TT)
18240     IF TT=1 THEN NM$="0"+NM$
18260     TT$=STR(NY):TT-LEN(TT$)-1:NY$=MID$(TT$,2,TT)
18280     TT$=STR$(NMO):TT-LEN(TT$)-1:NMO$=MID$(TT$,2,TT)
18300     IF TT=1 THEN NMO$="0"+NMO$
18320     TT$=STR$(ND):TT-LEN(TT$)-1:ND$=MID$(TT$,2,TT)
18340     IF TT=1 THEN ND$="0"+ND$
18360     LSET NTIME$=NH$+":"+NM$+":"+MID$(CTM$,7,2)
18380     LSET NDATE$=NY$+"-"+NMO$+"-"+ND$
18400     PUT #2, (LOF(2)/LAST(2))+1

18440     ;Creating datafile
18480     OPEN FFILE$ AS #4 LEN=48
18500     FILED #4, 10 AS DDATE$, 8 AS DTIME$, 10 AS DHRS$, 10 AS DWEIGHT$, 10
          AS DCHW$
18520     LSET DDATE$=CDT$
18540     LSET DTIME$=CTM$
18560     RSET DHRS$=STR$(0)
18580     RSET DWEIGHT$(STR$(IWEIGHT#)
18600     RSET DCHW$="0"
18620     PUT #4, LOF(4)/48+1
18640     CLOSE #4

18680     ;Storing sample specifications
18720     OPEN "SPECS.DAT: AS #4 LEN=125
18740     FIELD #4, 25 AS SD$, 15 AS SR$, 15 AS SF$, 15 AS SC$, 10 AS ST$, 10
          AS STH$, 10 AS SA$, 25 AS SCOND$
18760     LSET SD$=DISC$
18780     LSET SR$=RESIN$
18800     LSET SF=FIBER$
18820     LSET SC$=CONF$
18840     LSET ST$=TEMP$
18860     LSET STH$=THICK$
18880     LSET SA$=STR$(AREA)
18900     LSET SCOND$=COND$
18920     PUT #4, BATH
18940     CLOSE #4
18960     RETURN

19000     ;Fill jar "ONE" with water, set startflag
19040     PRINT:PRINT "SPECIMEN JAR IS BEING PREPARED AT THE MONENT, AT THE
          BEEP INTRODUCE THE SPECIMEN (1 HR.)"
19080     STARTBATH=ONE
19100     STARTFLAG=1
19120     VARNAME$="BATH.LOCATION"
```

```
19140   VARVALUE=STARTBATH
19160   GOSUB 6000
19180   GOSUB 7120; Get and uncap jar
19200   PROGNAME$="GET.JAR.FROM.CAPPING.NO.CAP":GOSUB 9060
19220   PROGNAME$="FILL.JAR":GOSUB 9060
19240   PROGNAME$="PUT.JAR.IN.CAPPING.NO.CAP":GOSUB 9060
19260   GOSUB 10460:GOSUB 8680; Cap jar and return to bath
19280   CTM$=TIME$
19300   ST=VAL(MID$(CTM,1,2))
19320   ST=ST+1
19340   TT$=STR$(ST):TT-LEN(TT$)-1:STARTTIME$=MID$(TT$,2,TT)
19360   IF TT=1 THEN STARTTIME$="0"+STARTTIME$
19380   STARTTIME$=STARTTIME$+MID$(CTM$,3,6)
19420   GET #3,STARTBATH
19440   RSET FU$="1"
19460   RSET WWO$="1"
19480   PUT #3,STARTBATH
19500   RETURN

19660   ;Sounding Beep for initalizing specimen waiting
19680   SOUND$="GGGGGGGGGG"
19700   PLAY ">XSOUND$;"
19720   BATH=STARTBATH
19740   GSOUB 16260
19760   STARTFLAG=0
19780   STARTBATH=0
19800   STARTTIME$="00:00:00"
19820   RETURN

19860   ;Search for jar with rating=2 for switching
19900   STARTBATH=TWO
19920   SWITCH=0
19960   ;Search for possible jar with rating=2 to switch sample with
19980   FOR C=1 TO (LOF(3)/LAST(3))
20000      GET #3,C
20020      JR=VAL(RATING$)
20040      IF JR<>2 GOTO 20120
20060      JFU=VAL(FU$)
20080      JWWO=VAL(WWO$)
20100      IF (JFU=0) AND (WWO=1) THEN SWITCH=C: GOTO 20240
20120   NEXT C
20140   PRINT "NO BATH LOCATIONS ARE AVAILABLE"
20160   STARTFLAG=0
20180   STARTBATH=0
20200   STARTTIME$="00:00:00"
20220   RETURN

20260   ;Switch jars
20300   VARNAME$="BATH.LOCATION"
20320   VARVALUE=STARTBATH
20340   GOSUB 6000
20360   GOSUB 7120; Get and uncap jar
20380   GOSUB 7440; Get specimen from jar
20400   PROGNAME$="SET.SAMPLE.ASIDE":GOSUB 9060
```

```
20420    PROGNAME$="GET.JAR.FROM.CAPPING.NO.CAP":GOSUB 9060
20440    PROGNAME$="EMPTY.JAR":GOSUB 9060
20460    PROGNAME$="FILL.JAR":GOSUB 9060
20480    PROGNAME$="PUT.JAR.IN.CAPPING.NO.CAP":GOSUB 9060
20500    GOSUB 10480:GOSUB 8680; Cap jar and return to bath
20520    STARTFLAG=1
20540    CTM$-TIME$
20560    VARNAME$="BATH.LOCATION"
20580    VARVALUE=SWITCH
20600    GOSUB 6000
10620    GOSUB 7120; Get and uncap jar
20640    PROGNAME$="GET.SAMPLE.FROM.JAR":GOSUB 9060; Orient jar
20660    GOSUB 7760; Get specimen from holder
20680    PROGNAME$="CLEAR.BEAM.SENSOR":GOSUB 9060
20700    GOSUB 8460; Put sample in jar
20720    GOSUB 8600; Cap and return jar

20780    ;Update 'STUDIES.DAT'
20820    FOR J=1 TO LOF(2)/LAS(2)
20840       GET #2,J
20860       IF BATH=STARTBATH GOTO 20900
20880    NEXT J
20900    RSET BATH$=SWITCH
20920    PUT #2,J

20960    ;Update 'DATAFILE.DAT'
21000    OPEN "DATAFILE.DAT" AS #4 LEN=12
21020    FIELD#4, 12 AS FILE$
21040    GET #4, STARTBATH
21060    FFILE$=FILE$
21080    RSET FILE$="EMPTY"
21100    PUT #4,SWITCH
21120    RSET FILE$=FFILE$
21140    PUT #4,SWITCH
21160    CLOSE #4

21200    ;Update 'JARSTATS.DAT'
21240    GET #3,STARTBATH
21260    JCI$=CI$
21280    RSET CI$="0000:00"
21300    PUT #3,STARTBATH
21320    GET #3,SWITCH
21340    RSET FU$="1"
21360    RSET CI$=JCI$
21380    PUT #3,SWITCH

21420    ;Update SPECS.DAT
21460    OPEN "SPECS.DAT" AS #4 LEN=125
21480    FIELD #4, 25 AS SD$, 15 AS SR$, 15 AS SF$, 15 AS SC$, 10 AS ST$, 10
         AS STH$, 10 AS SA$, 25 AS SCOND$
21500    GET #4,STARTBATH
21520    PUT #4,SWITCH
21540    GET #4,3
21560    PUT #4,STARTBATH
```

```
21580    CLOSE #4
21600    RETURN
```

```
21640    ;JARSTATS.DAT set-up (Function key 6)
21700    LOOP=0
21720    IF LOF(3)/13>0 THEN GOTO 22260
21740    FOR J=1 TO 11
21760       RSET RATING$="1"
21780       RSET FU$="0"
21800       LSET CI$="0000:00"
21820       RSET WWO$="0"
21840       PUT #3,J
21860    NEXT J
21880    GET #3,3
21900    RSET RATING$="0"
21920    PUT #3,3
21940    GET #3,4
21960    RSET RATING$="0"
21980    PUT #3,4
22000    FOR J=12 TO 20
22020       RSET RATING$="2"
22040       RSET FU$="0"
22060       RSET CI$="0000:00"
22080       RSET WWO$="0"
22100       PUT #3,J
22120    NEXT J
22120    FOR J=21 TO 30
22140       RSET RATING$="0"
22160       RSET FU$="0"
22180       RSET CI$="0000:00"
22200       RSET WWO$="0"
22240       PUT #3,J:NEXT J
22260    PRINT:PRINT "SET UP FILE JARSTATS.DAT"
22280    PRINT:PRINT "INFORMATION REQUIRED IS AS FOLLOS:"
22300    PRINT:PRINT "RATING: 0 - UNUSABLE"
22320    PRINT "          1 - < 24 HOURS"
22340    PRINT "          2 - >= 24 HOURS"
22360    PRINT:PRINT "USAGE: 1 - IN USE; 2 - NOT IN USE"
22380    PRINT:PRINT "TIME INTERVAL: XXXX:XX (HRS:MIN)"
22400    PRINT:PRINT "FILLED/NOT FILLED: 1-FILLED; 0-NOT FILLED"
22420    PRINT:INPUT :SETUP ALL BATH LOCATIONS? Y/N ",ANS$
22440    IF MID$(ANS$,1,1)="Y" GOTO 22780
22460    PRINT:INPUT "ENTER BATH LOCATION", ANS$
22480    IF ANS$="" GOTO 22920
22500    BATH=VAL(ANS$)
22520    GET #3,BATH
22540    PRINT:PRINT "RATING= ",RATING$
22560    PRINT:PRINT "USAGE= ",FU$
22580    PRINT:PRINT "INTERVAL= ",CI$
22600    PRINT:PRINT "FILLED/NOT FILLED= ",WWO$
22620    PRINT:INPUT "ENTER PARAMETERS (4): ",JR,JFU,JCI$,JWWO
22640    RSET RATING$=STR$(JR)
22660    RSET FU$=STR$(FJFU)
22680    RSET CI$=JCI$
22700    RSET WWO$=STR$(JWWO)
22720    PUT #3,BATH
22740    GOTO 22460
```

```
22780    FOR J=1 TO 30
22800       PRINT:PRINT "ENTER WWO$ PARAMETER FOR BATH LOCATION ",J
22820       INPUT JR
22840       GET #3,J
22860       RSET WWO$=STR$(JR)
22880       PUT #3,J
22900    NEXT J
22920    RETURN
```

```
22940     ;Print status report and delete file (Function key 4)
23040     LPRINT "  BATH          TYPE OF          ERROR"
23060     LPRINT "                 ERROR"
23080     LPRINT
23100     FOR ST=1 TO LOF(5)/LAST(5)
23120        GET #5,ST
23140        LPRINT "     ",B$;
23160        LPRINT "             ";
23180        TYPE=VAL(TYPE$)
23200        IF TYPE=0 THEN LPRINT "NOT FATAL      ";
23220        IF TYPE=1 THEN LPRINT "FATAL          ";
23240        VAL=LEN(MESS$)
23260        IF VAL<=50 THE LPRINT MESS$:GOTO 23340
23280        MESSAGE$=MID$(MESS$,1,50):LPRINT MESSAGE$
23300        LPRINT "                           ";
23320        LPRINT MID$(,51,VAL-50)
23340     NEXT ST
23360     CLOSE #5
23380     OPEN "STATUS.REP" FOR OUTPUT AS #5
23400     CLOSE #5
23420     RETURN 460
```

DISTRIBUTION LIST

No. of Copies	To
1	Office of the Under Secretary of Defense for Research and Engineering, The Pentagon, Washington, DC 20301
	Commander, U.S. Army Laboratory Command, 2800 Powder Mill Road, Adelphi, MD 20783-1145
1	ATTN: AMSLC-IM-TL
	Commander, Defense Technical Information Center, Cameron Station, Building 5, 5010 Duke Street, Alexandria, VA 22304-6145
2	ATTN: DTIC-FDAC
1	Metals and Ceramics Information Center, Battelle Columbus Laboratories, 505 King Avenue, Columbus, OH 43201
	Commander, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709-2211
1	ATTN: Information Processing Office
	Commander, U.S. Army Materiel Command, 5001 Eisenhower Avenue, Alexandria, VA 22333
1	ATTN: AMCLD
	Commander, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD 21005
1	ATTN: AMXSY-MP, H. Cohen
	Commander, U.S. Army Electronics Research and Development Command, Fort Monmouth, NJ 07703-5000
1	ATTN: AMDSD-L
1	AMDSD-E
	Commander, U.S. Army Missile Command, Redstone Scientific Information Center, Redstone Arsenal, AL 35898-5241
1	ATTN: AMSMI-RD-CS-R/Doc
1	AMSMI-RLM
	Commander, U.S. Army Armament, Munitions and Chemical Command, Dover, NJ 07801
2	ATTN: Technical Library
1	AMDAR-LCA, Mr. Harry E. Pebly, Jr., PLASTEC, Director
	Commander, U.S. Army Natick Research, Development, and Engineering Center, Natick, MA 01760
1	ATTN: Technical Library
	Commander, U.S. Army Satellite Communications Agency, Fort Monmouth, NJ 07703
1	ATTN: Technical Document Center
	Commander, U.S. Army Tank-Automotive Command, Warren, MI 48397-5000
1	ATTN: AMSTA-ZSK
2	AMSTA-TSL, Technical Library

No. of Copies	To
	Commander, White Sands Missile Range, NM 88002
1	ATTN: STEWS-WS-VT
	President, Airborne, Electronics and Special Warfare Board, Fort Bragg, NC 28307
1	ATTN: Library
	Director, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD 21005
1	ATTN: SLCBR-TSB-S (STINFO)
	Commander, Dugway Proving Ground, Dugway, UT 84022
1	ATTN: Technical Library, Technical Information Division
	Commander, Harry Diamond Laboratories, 2800 Powder Mill Road, Adelphi, MD 20783
1	ATTN: Technical Information Office
	Director, Benet Weapons Laboratory, LCWSL, USA AMCCOM, Watervliet, NY 12189
1	ATTN: AMSMC-LCB-TL
1	AMSMC-LCB-R
1	AMSMC-LCB-RM
1	AMSMC-LCB-RP
	Commander, U.S. Army Foreign Science and Technology Center, 220 7th Street, N.E., Charlottesville, VA 22901
1	ATTN: Military Tech
	Commander, U.S. Army Aeromedical Research Unit, P.O. Box 577, Fort Rucker, AL 36360
1	ATTN: Technical Library
	Director, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, VA 23604-5577
1	ATTN: SAVDL-E-MOS (AVSCOM)
	U.S. Army Aviation Training Library, Fort Rucker, AL 36360
1	ATTN: Building 5906-5907
	Commander, U.S. Army Agency for Aviation Safety, Fort Rucker, AL 36362
1	ATTN: Technical Library
	Commander, USACDC Air Defense Agency, Fort Bliss, TX 79916
1	ATTN: Technical Library
	Commander, U.S. Army Engineer School, Fort Belvoir, VA 22060
1	ATTN: Library
	Commander, U.S. Army Engineer Waterways Experiment Station, P. O. Box 631, Vicksburg, MS 39180
1	ATTN: Research Center Library

No. of Copies	To
	Commandant, U.S. Army Quartermaster School, Fort Lee, VA 23801
1	ATTN: Quartermaster School Library
	Naval Research Laboratory, Washington, DC 20375
1	ATTN: Code 5830
2	Dr. G. R. Yoder - Code 6384
	Chief of Naval Research, Arlington, VA 22217
1	ATTN: Code 471
1	Edward J. Morrissey, AFWAL/MLTE, Wright-Patterson Air Force, Base, OH 45433
	Commander, U.S. Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, OH 45433
1	ATTN: AFWAL/MLC
1	AFWAL/MLLP, M. Forney, Jr.
1	AFWAL/MLBC, Mr. Stanley Schulman
	National Aeronautics and Space Administration, Marshall Space Flight Center, Huntsville, AL 35812
1	ATTN: R. J. Schwinghammer, EH01, Dir, M&P Lab
1	Mr. W. A. Wilson, EH41, Bldg. 4612
	U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD 20899
1	ATTN: Stephen M. Hsu, Chief, Ceramics Division, Institute for Materials Science and Engineering
1	Committee on Marine Structures, Marine Board, National Research Council, 2101 Constitution Ave., N.W., Washington, DC 20418
1	Librarian, Materials Sciences Corporation, Guynedd Plaza 11, Bethlehem Pike, Spring House, PA 19477
1	The Charles Stark Draper Laboratory, 68 Albany Street, Cambridge, MA 02139
	Wyman-Gordon Company, Worcester, MA 01601
1	ATTN: Technical Library
	Lockheed-Georgia Company, 86 South Cobb Drive, Marietta, GA 30063
1	ATTN: Materials and Processes Engineering Dept. 71-11, Zone 54
	General Dynamics, Convair Aerospace Division, P.O. Box 748, Fort Worth, TX 76101
1	ATTN: Mfg. Engineering Technical Library
1	Mechanical Properties Data Center, Belfour Stulen Inc., 13917 W. Bay Shore Drive, Traverse City, MI 49684
1	Mr. R. J. Zentner, EAI Corporation, 626 Towne Center Drive, Suite 205, Joppatowne, MD 21085-4440
	Director, U.S. Army Materials Technology Laboratory, Watertown, MA 02172-0001
2	ATTN: SLCMT-TML
1	Author

U.S. Army Materials Technology Laboratory
Watertown, Massachusetts 02172-0001
AUTOMATED IMMERSION TESTING ROBOTIC SYSTEM - HARDWARE AND SOFTWARE DESIGN
Suzanne G. W. Dunn

AD________________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Laboratory automation
Immersion testing
Robotics

Technical Report MTL TR 89-36, April 1989, 59 pp-illus-table, D/A Project: 1L162105.AH84

Hardware and software development of a fully automated robotic work cell to determine the water sorption-diffusion properties of plastics and organic matrix composite materials and to handle all data acquisition is described. Immersion testing involves measuring weight changes of specimens as a function of the time they are immersed in water at a particular temperature. This test method is labor intensive and tedious, involving repetitive operations such as transferring liquids and specimens, blotting and weighing specimens, and recording data. Immersion studies can take years to complete and generate enormous amounts of data which must be recorded and evaluated. Through automation, immersion testing is no longer limited to an 8-hour work schedule, labor costs have been reduced, productivity has been increased tenfold, and the reliability, precision, and accuracy of the test have been improved.

U.S. Army Materials Technology Laboratory
Watertown, Massachusetts 02172-0001
AUTOMATED IMMERSION TESTING ROBOTIC SYSTEM - HARDWARE AND SOFTWARE DESIGN
Suzanne G. W. Dunn

AD________________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Laboratory automation
Immersion testing
Robotics

Technical Report MTL TR 89-36, April 1989, 59 pp-illus-table, D/A Project: 1L162105.AH84

Hardware and software development of a fully automated robotic work cell to determine the water sorption-diffusion properties of plastics and organic matrix composite materials and to handle all data acquisition is described. Immersion testing involves measuring weight changes of specimens as a function of the time they are immersed in water at a particular temperature. This test method is labor intensive and tedious, involving repetitive operations such as transferring liquids and specimens, blotting and weighing specimens, and recording data. Immersion studies can take years to complete and generate enormous amounts of data which must be recorded and evaluated. Through automation, immersion testing is no longer limited to an 8-hour work schedule, labor costs have been reduced, productivity has been increased tenfold, and the reliability, precision, and accuracy of the test have been improved.

U.S. Army Materials Technology Laboratory
Watertown, Massachusetts 02172-0001
AUTOMATED IMMERSION TESTING ROBOTIC SYSTEM - HARDWARE AND SOFTWARE DESIGN
Suzanne G. W. Dunn

AD________________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Laboratory automation
Immersion testing
Robotics

Technical Report MTL TR 89-36, April 1989, 59 pp-illus-table, D/A Project: 1L162105.AH84

Hardware and software development of a fully automated robotic work cell to determine the water sorption-diffusion properties of plastics and organic matrix composite materials and to handle all data acquisition is described. Immersion testing involves measuring weight changes of specimens as a function of the time they are immersed in water at a particular temperature. This test method is labor intensive and tedious, involving repetitive operations such as transferring liquids and specimens, blotting and weighing specimens, and recording data. Immersion studies can take years to complete and generate enormous amounts of data which must be recorded and evaluated. Through automation, immersion testing is no longer limited to an 8-hour work schedule, labor costs have been reduced, productivity has been increased tenfold, and the reliability, precision, and accuracy of the test have been improved.

U.S. Army Materials Technology Laboratory
Watertown, Massachusetts 02172-0001
AUTOMATED IMMERSION TESTING ROBOTIC SYSTEM - HARDWARE AND SOFTWARE DESIGN
Suzanne G. W. Dunn

AD________________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Laboratory automation
Immersion testing
Robotics

Technical Report MTL TR 89-36, April 1989, 59 pp-illus-table, D/A Project: 1L162105.AH84

Hardware and software development of a fully automated robotic work cell to determine the water sorption-diffusion properties of plastics and organic matrix composite materials and to handle all data acquisition is described. Immersion testing involves measuring weight changes of specimens as a function of the time they are immersed in water at a particular temperature. This test method is labor intensive and tedious, involving repetitive operations such as transferring liquids and specimens, blotting and weighing specimens, and recording data. Immersion studies can take years to complete and generate enormous amounts of data which must be recorded and evaluated. Through automation, immersion testing is no longer limited to an 8-hour work schedule, labor costs have been reduced, productivity has been increased tenfold, and the reliability, precision, and accuracy of the test have been improved.